



CHAPTER 4

Formal Requirements

Topics:

- 4.1 Analysis Model
- 4.2 Analysis Model Objectives
 - Elements of Analysis Model
- 4.3 Data Model
- 4.4 Creating an Entity/Relationship Model
- 4.5 Creating a Data Flow Model



1.4 Analysis Model

The analysis model, actually a set of models, is the first technical representation of a system. Over the years many methods have been proposed for analysis modeling. However, two now dominate. The first, **structured analysis**, is a classical modeling method and is described in this chapter. The other approach, **object oriented analysis**.

نموذج التحليل (*analysis model*) هو مجموعة من النماذج (models)، وهو التمثيل الفني (التقني) الأول للنظام. وعلى مدى السنوات ، تم اقتراح العديد من الأساليب لنمذجة التحليل (*analysis model*). , واهم طريقتين مستخدمة هما اولا : التحليل الهيكلي (structured analysis) و هو اسلوب النمذجة الكلاسيكية (التقليدية) وسوف نتناولها في هذا الفصل. ثانيا : التحليل الشيئي (object oriented analysis).

التحليل الهيكلي (Structured Analysis) :

هو عملية تهدف لدراسة النظام بغية معرفة طريقة أدائه لعمله والمشاكل التي تعترضه وتحديد المتطلبات وذلك باستخدام منهجية التفكير والتجزئة الهيكلية (الهرمية) (Structured Methodology) باستخدام ادوات التحليل الهيكلية للنظام الحالي والمقترح.

التحليل الشيئي (object oriented analysis) :

هو طريقة للتعامل مع المتطلبات من وجهة نظر الفصائل (Classes) و الكائنات (Objects).

4.2 Analysis Model Objectives:

The analysis model must achieve three primary objectives:

- (1) to describe what the customer requires,
- (2) to establish a basis for the creation of a software design, and
- (3) to define a set of requirements that can be validated once the software is built. To accomplish these objectives, the analysis model derived during structured analysis takes the form illustrated in Figure (4.1)

اهداف نموذج التحليل (Analysis Model Objectives)

نموذج التحليل يجب تحقيقها بثلاثة أهداف رئيسية هي:

- (1) يقوم بوصف مطالب الزبون
- (2) بناء قاعدة لعملية تكوين أو تصميم الـ Software
- (3) تعريف مجموعة من المتطلبات Requirements التي يجب على الـ Software حفظها متى ما تم بناءها.

ولتحقيق هذه الأهداف، نموذج التحليل (*analysis model*) المتحقق خلال التحليل الهيكلي (structured analysis) يأخذ الشكل الموضح في المخطط (4.1).

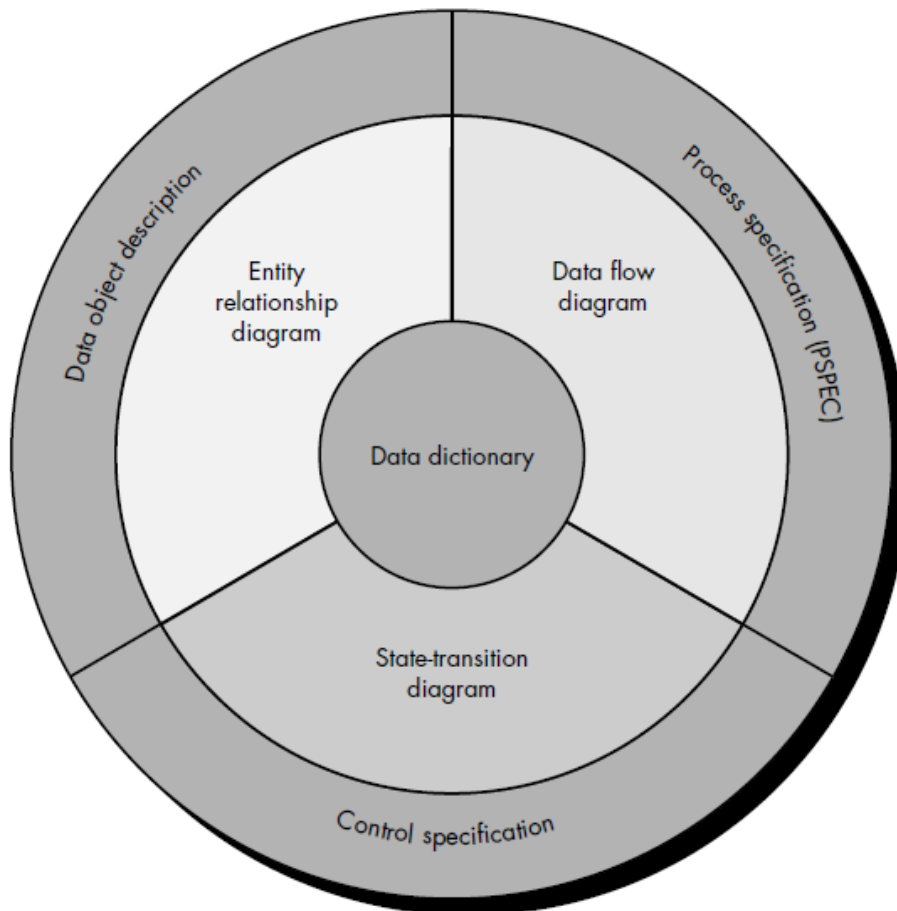


Figure 4.1: The structure of the analysis model

Elements of Analysis Model (Components for structure of the analysis model)

- 1- Data dictionary
- 2- Entity relation diagram (ERD)
- 3- Data flow diagram (DFD)
- 4- State transition diagram (STD)

At the core of the model lies the **data dictionary**: is an organized listing of all data elements that are pertinent to the system, with precise, rigorous definitions so that both user and system analyst will have a common understanding of inputs, outputs, components of stores and [even] intermediate calculations.

Or **data dictionary** a repository that contains descriptions of all data objects consumed or produced by the software.



Three different diagrams surround the core:

1- The ***entity relationship diagram (ERD)*** depicts relationships between data objects. The ERD is the notation that is used to conduct the data modeling activity. The attributes of each data object noted in the ERD can be described using a data object description.

2- The ***data flow diagram (DFD)*** : is a graphical representation that depicts information flow and the transforms that are applied as data move from input to output. The basic form of a data flow diagram, **also known as a data flow graph or a bubble chart**.

serves two purposes: (1) to provide an indication of how data are transformed as they move through the system and (2) to depict the functions (and sub functions) that transform the data flow. The DFD provides additional information that is used during the analysis of the information domain and serves as a basis for the ***modeling of function***. **A description of each function presented in the DFD is contained in a process specification (PSPEC)**.

DFD are used to represent data and the processes that manipulate it.

3- ***State transition diagram (STD)***): By studying the STD, a software engineer can determine the behavior of the system and, more important, can ascertain whether there are "holes" in the specified behavior. STD is a behavioral model

The STD indicates how the system behaves as a consequence of external events. To accomplish this, the STD represents the various modes of behavior (called *states*) of the system and the manner in which transitions are made from state to state. The STD serves as the basis for behavioral modeling. **Additional information about the control aspects of the software is contained in the control specification (CSPEC)**.

Data object Description: The attributes of each data object noted in the ERD can be described using a data object description.

In other word: incorporates the data object and all of its attributes.

Process Specification (PSPEC): is used to describe all flow model processes that appear at the final level of refinement. The content of the process specification can include narrative text, a *program design language* (PDL) description of the process algorithm, mathematical equations, tables, diagrams, or charts.

The control specification (CSPEC): represents the behavior of the system in two different ways. The CSPEC contains a state transition diagram that is a sequential specification of behavior. It can also contain a program activation table—a combinatorial specification of behavior.



CSPEC is used to indicate (1) how the software behaves when an event or control signal is sensed and (2) which processes are invoked as a consequence of the occurrence of the event.

عناصر نموذج التحليل :

Elements of Analysis Model (Components for structure of the analysis model)

- 1- Data dictionary
- 2- Entity relation diagram (ERD)
- 3- Data flow diagram (DFD)
- 4- State transition diagram (STD)

في مركز النموذج يكمن قاموس البيانات (*data dictionary*) : قائمة او مستودع لكافة عناصر البيانات (كائنات) (*data objects*) الخاصة بالمنظومة او وصف لمخازن البيانات وانسيابها والموجودة في مخطط DFD .

- او مستودع يحتوي على وصف لكافة الكائنات البيانات المستهلكة أو المنتجة من قبل البرنامج.

- هناك ثلاثة مخططات مختلفة تحيط المركز (data dictionary) :

- 1- مخطط الكائنات العلائقية ((*entity relation diagram* (ERD))
- 2- مخطط انسياب البيانات (تدفق البيانات) ((*data flow diagram* (DFD))
- 3- مخطط انتقال الحالة ((*state transition diagram* (STD))

1- مخطط الكائنات العلائقية ((*entity relation diagram* (ERD)) :

يعتبر مخطط الكائنات العلائقية ERD (Entity Relationship Diagram) بداية صحيحة من قبل محلل النظم لفهم متطلبات البيانات Data Requirements الخاصة بالمنظومة تحت الاعداد. ويتكون مخطط الكائنات العلائقية من اشكال هندسية تشبه المخطط الانسيابي تبين الكائنات data object والعلاقة Relationships بين هذه الكائنات وايضا الخصائص Attributes لكل كائن أو علاقة. ويعتبر هذا المخطط أيضا أساسا لتصميم قاعدة البيانات .

2- مخطط انسياب البيانات (تدفق البيانات) ((*Data Flow Diagram* (DFD)) :

وهو مخطط هيكل رسومي يبين صورة لحركة انسياب البيانات داخل النظام بين مخازن البيانات والمعالجة والكيانات الخارجية . إن الشكل الأساسي للـ DFD يعرف باسم (Flow graph أو Bubble chart) الرسم البياني لتدفق البيانات أو مخطط الفقاعة.

يستخدم لغرضين : اولاً ، يعطينا مؤشرات حول كيفية انتقال البيانات خلال النظام. ثانياً، يوضح الدوال التي تقوم بتناقل البيانات أو تحويل البيانات.

3- مخطط انتقال الحالة ((*State Transition Diagram* (STD)) :

من خلال دراسة STD يستطيع مهندس البرمجيات من تحديد سلوك النظام، والأهم، يمكن التأكد مما إذا كانت هناك ثغرات في سلوك محدد.

ويشير STD الى الطريقة التي يتصرف بها النظام نتيجة لأحداث خارجية. ولتحقيق ذلك، يمثل STD مختلف أنماط السلوك (تدعى حالات states) للنظام وطريقة الانتقال من حالة إلى أخرى. يقدم STD كأساس للنماذج السلوكية (behavioral modeling).

Data object Description: هي عملية وضع الخصائص أو الصفات attributes لكل data object.



Process SPECification (PSPEC): يستخدم لوصف جميع عمليات نموذج التدفق (flow model) التي تظهر في المستوى النهائي من التحسين (التنقية). و PSPEC يمكن أن تشمل النص السردى، ووصف لغة تصميم البرنامج (PDL)، والمعادلات الرياضية والجداول والرسوم البيانية، أو المخططات.

:Control Specification (CSPEC)

مواصفات التحكم (CSPEC): تمثل سلوك النظام بطريقتين مختلفتين، CSPEC يحتوي على مخطط انتقال الحالة (STD) التي هي مواصفات متتابعة من السلوك. ويمكن أن تحتوي أيضا على جدول تفعيل البرنامج (مواصفات اندماج السلوك).

Control Flow Diagram (CFD): contains the same processes as the DFD, but shows control flow, rather than data flow. CFD show how events flow among processes and illustrate those external events that cause various processes to be activated.

Control Flow Diagram (CFD) (مخطط التحكم في التدفق): يحتوي على نفس العمليات في DFD، ولكن يظهر التحكم في التدفق (control flow)، بدلا من تدفق البيانات (data flow). و CFD يبين كيفية تدفق الأحداث بين العمليات ويوضح (يشرح) الأحداث الخارجية التي تتسبب في عمليات مختلفة لتفعيلها.

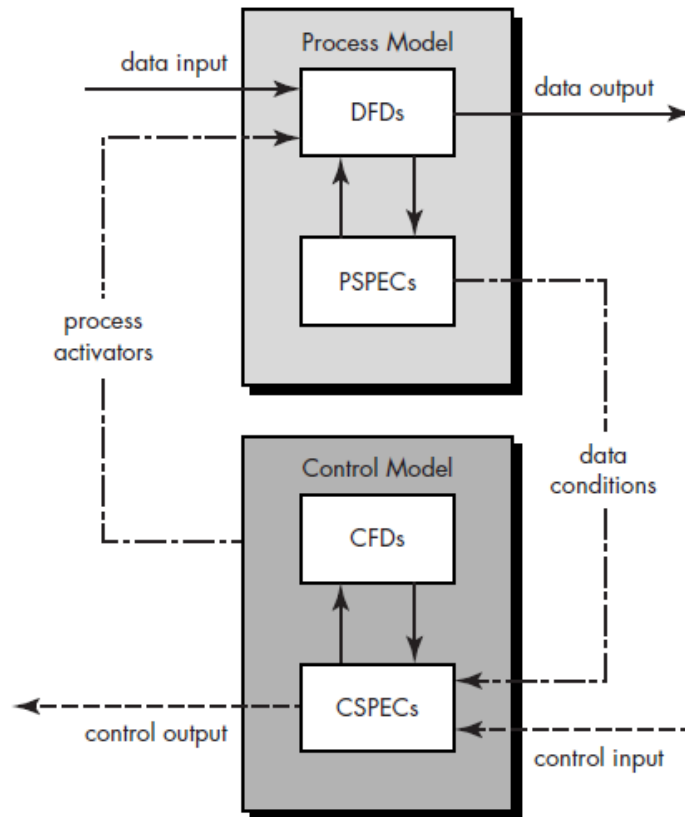


Figure 4.2:
The relationship
between data model
and control model



4.3 Data Model :

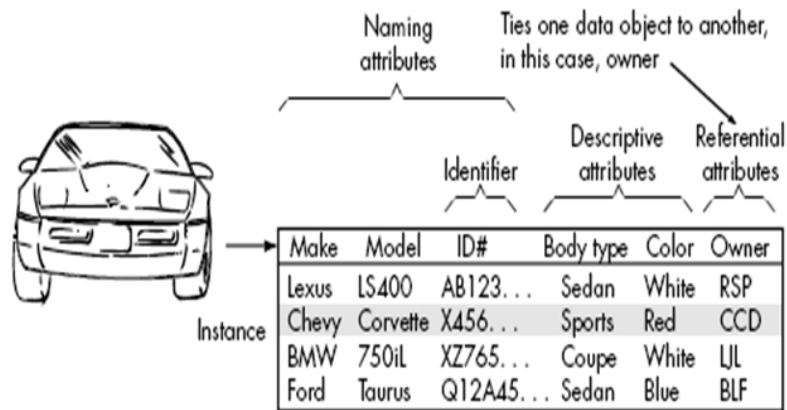
The data model consists of three interrelated pieces of information:

- 1- **Data object** (input and output from a system)
- 2- **Attributes** that describe the data object
- 3- **Relationships** that connect data objects to one another.

1- Data object : A data object is a representation of almost any composite information that must be understood by software. By composite information, we mean something that has a number of different properties or attributes. A data object can be an external entity (e.g., anything that produces or consumes information), a thing (e.g., a report or a display), an occurrence (e.g., a telephone call) or event (e.g., an alarm), a role (e.g., salesperson), an organizational unit (e.g., accounting department), a place (e.g., a warehouse), or a structure (e.g., a file). A data object encapsulates data only—there is no reference within a data object to operations that act on the data.

2- Attributes: Attributes define the properties of a data object and take on one of three different characteristics. They can be used to (1) name an instance of the data object, (2) describe the instance, or (3) make reference to another instance in another table.

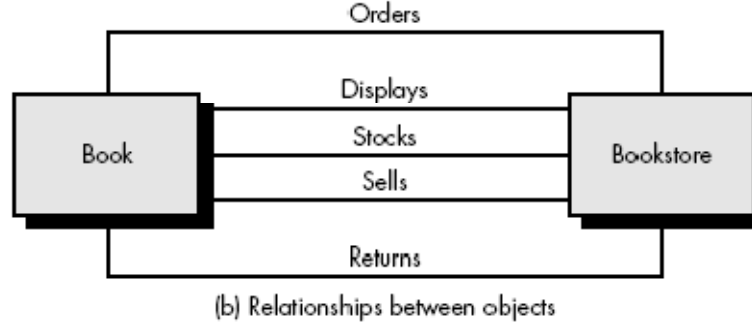
- In addition, one or more of the attributes must be defined as an identifier—that is, the identifier attribute becomes a "key" when we want to find an instance of the data object. In some cases, values for the identifier(s) are unique, although this is not a requirement.



3- Relationships: Data objects are connected to one another in different ways. Relationships define a set of object/relationship pairs that define the relevant relationships. It is important to note that object/relationship pairs are bidirectional. That is, they can be read in either direction. A bookstore orders books or books are ordered by a bookstore.

There are two basic points in the Relationships be aware of:

- 1- **Cardinality**
- 2- **Modality**



1- Cardinality: the specification of the number of occurrences of one [object] that can be related to the number of occurrences of another [object]. Cardinality is usually expressed as simply 'one' or 'many.'

Taking into consideration all combinations of 'one' and 'many,' two [objects] can be related as :

- **One-to-one (1:1)**—An occurrence of [object] 'A' can relate to one and only one occurrence of [object] 'B,' and an occurrence of 'B' can relate to only one occurrence of 'A.'
- **One-to-many (1:N)**—One occurrence of [object] 'A' can relate to one or many occurrences of [object] 'B,' but an occurrence of 'B' can relate to only one occurrence of 'A.' For example, a mother can have many children, but a child can have only one mother.
- **Many-to-many (M:N)**—An occurrence of [object] 'A' can relate to one or more occurrences of 'B,' while an occurrence of 'B' can relate to one or more occurrences of 'A.' For example, an uncle can have many nephews, while a nephew can have many uncles.

2- Modality: The *modality* of a relationship is 0 if there is no explicit need for the relationship to occur or the relationship is optional. The modality is 1 if an occurrence of the relationship is mandatory.

- **نمذجة البيانات (Data Model) :** تحتوي على ثلاث قطع مترابطة من المعلومات :
- 1- كائن البيانات Data object : (المدخلات والمخرجات من النظام)
 - 2- الصفات Attributes : التي تصف كائن البيانات
 - 3- العلاقات Relationships : التي تربط كائنات البيانات إلى بعضها البعض.

Data Object : هو تمثيل لأي مجموعة من المعلومات تلك التي يجب على الـ Software أن يتفهمها . نعني بمجموعة المعلومات شيء ما يحتوي على عدد مختلف من الخصائص أو الصفات . ممكن أن يكون الـ Data Object هو كيان خارجي (مثلا أي شيء يستهلك أو ينتج معلومات) ، تواجد (مثلا تقرير report) ، حدث (مثلا إنذار) ، دور (الشخص البائع) ، مكان (مخزن depository) ، هيكل (ملف File) .
إن الـ Data Object يغلف البيانات فقط (يشفر البيانات) Encapsulate حيث لا يوجد إشارة داخل الـ Object عن ماهية العمليات التي ممكن أن تعمل على تلك البيانات .



Attributes: تعرّف الصفات خصائص الـ Data Object الذي يأخذ واحد من الفروع الثلاثة :-

1. اسم الـ Data Object .
2. وصف الـ Data Object .
3. إشارة إلى Data Object آخر .

Relationships: أن الـ Data Object مترابطة احدها مع الآخر عن طريق اتجاهات مختلفة ، هنالك نقطتين أساسيتين بالـ Relationships يجب الانتباه لها: 1- Cardinality 2- Modality

حيث إن علاقة الـ Data Object الواحد بالآخر لا تعطي مهندس البرمجيات معلومات كافية عن العلاقة.

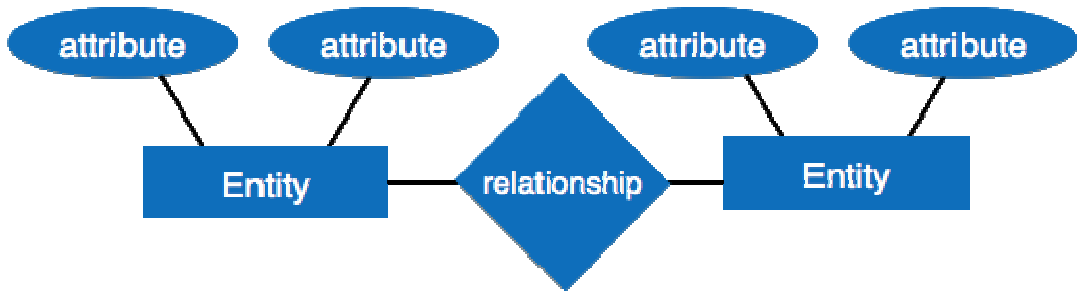
Cardinality: هي عملية تمثيل لعدد مرات تكرار ظهور الـ Object في علاقة وتأخذ قيمتين (many or one). مع الأخذ بعين الاعتبار جميع الارتباطات من "one" و "many"، بين كائنين (two objects) يمكن أن تكون العلاقة على النحو التالي:

1. one to one (1 : 1)
2. one to many (1 : n)
3. many to many (m : n)

Modality: هي صفر إذا لم تكن هنالك حاجة للعلاقة (اختيارية: optional) و واحد إذا كانت العلاقة مُوجبة (الزامي: يجب).

4.4 Creating an Entity/Relationship Diagram

The entity/relationship diagram enables a software engineer to fully specify the data objects that are input and output from a system, the attributes that define the properties of these objects, and their relationships. Like most elements of the analysis model, the ERD is constructed in an iterative manner. The following approach is taken:



key : مفتاح
attribute: خاصية



امثلة على مخطط الكائنات العلائقية (ERD)

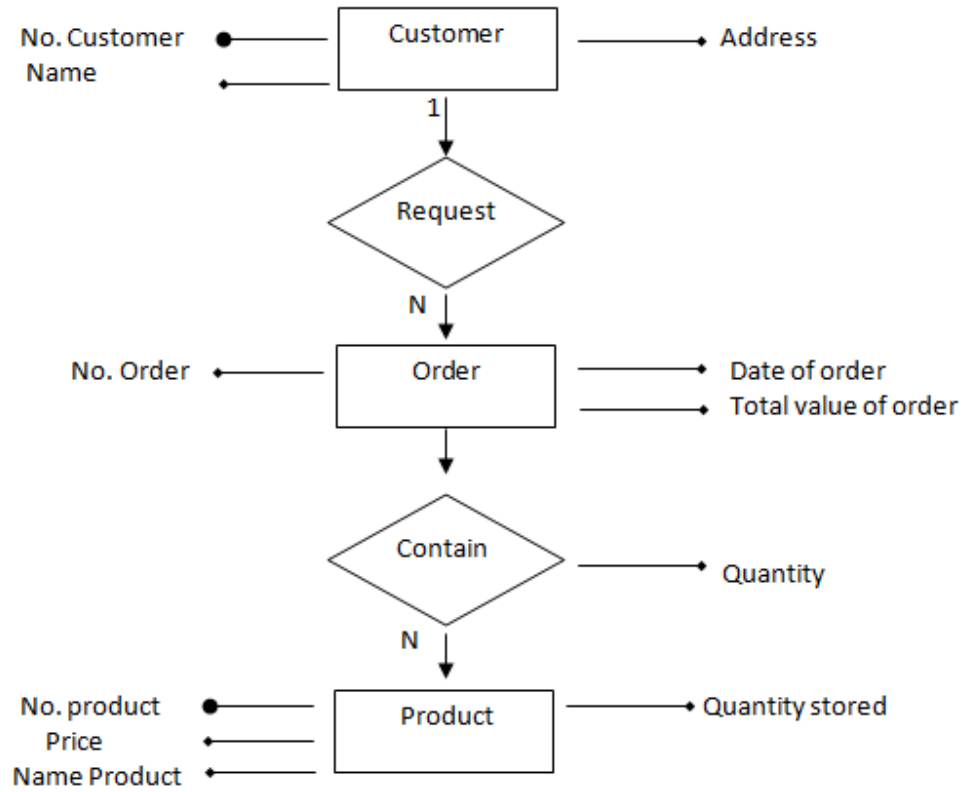


Figure: ERD FOR Sales System

Sales System: منظومة مبيعات
Customer: الزبون
No. Customer: اسم الزبون
Name: الاسم
Address: العنوان
Request: طلب
Order: طلبية
No. Order: رقم الطلبية
Date of order: تاريخ الطلبية
Total value of order: مجموع قيمة الطلبية
Contain: تحتوي
Quantity: الكمية
Quantity stored: الكمية المخزونة
Product: المنتج
No. product: رقم المنتج
Name Product: اسم المنتج, Price: السعر

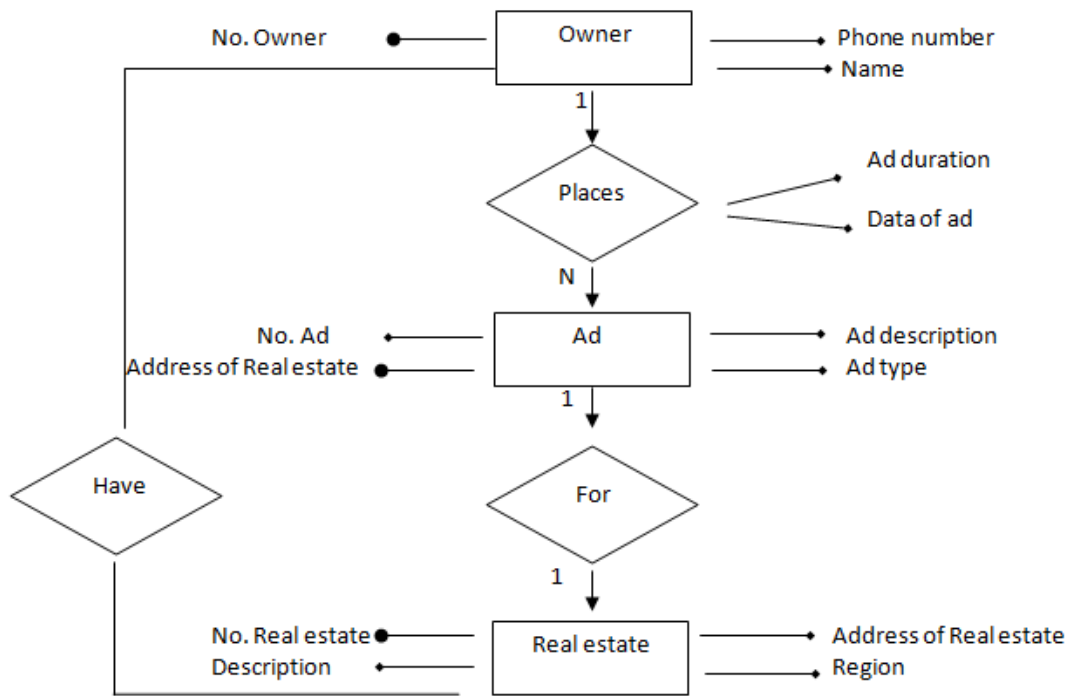


Figure: ERD FOR Real estates System

Real estates System:	منظومة عقارات
Owner:	المالك
No. Owner:	اسم الزبون
Name:	الاسم
Address:	العنوان
Phone number:	رقم الهاتف
places:	يضع
Advertisement (ad):	اعلان
Ad duration :	مدة الاعلان
Data of ad :	تاريخ وضع الاعلان
Ad description :	وصف الاعلان
Ad type :	نوع الاعلان
No.ad :	رقم الاعلان
Real estate :	العقار
Address of Real estate:	عنوان العقار
Have :	يملك
For :	يخص
Region :	المنطقة
No. Real estate:	رقم العقار
Description :	الوصف

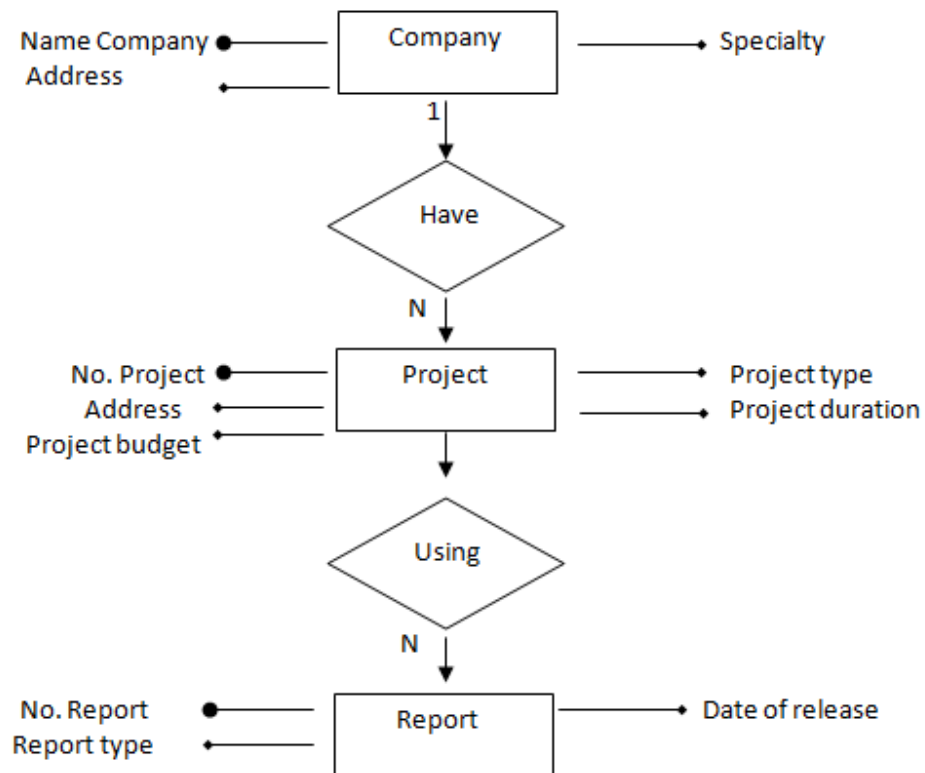


Figure: ERD for build System

Build System: منظومة بناء
 Company: شركة
 Name Company: اسم الشركة
 Address: العنوان
 Specialty: التخصص
 Have: تمتلك
 Project: مشروع
 No. Project: رقم المشروع
 Project budget: الميزانية
 Project type: نوع المشروع
 Project duration: مدة المشروع
 Using: يستخدم
 Report: تقرير
 No. Report: رقم التقرير
 Report type: نوع التقرير
 Date of release: تاريخ الاصدار



4.5 Creating a Data Flow Model

A few simple guidelines can aid immeasurably during derivation of a data flow diagram:

- (1) the level 0 data flow diagram should depict the software/system as a single bubble;
- (2) primary input and output should be carefully noted;
- (3) Refinement should begin by isolating candidate processes, data objects, and stores to be represented at the next level;
- (4) All arrows and bubbles should be labeled with meaningful names;
- (5) Information flow continuity must be maintained from level to level, and
- (6) One bubble at a time should be refined. There is a natural tendency to overcomplicate the data flow diagram.

خلق مخطط انسياب البيانات (Creating a Data Flow Model) :

هنالك مجموعة بسيطة من الخطوط العريضة التي تساعدنا في عملية اشتقاق أو تكوين data flow diagram وهي :

- 1- إن المستوى صفر لـ DFD يجب أن يوضح النظام أو الـ Software بفقاعة مفردة .
- 2- إن المدخلات و المخرجات الأساسية يجب أن يتم تحسينها بعناية .
- 3- إن عملية التصفيه refinement يجب أن تبدأ بعزل الـ Process أو Function ، data object و الـ data store لكي يتم تمثيلها في المستوى التالي.
- 4- جميع الأسهم والفقااعات يجب أن يتم إعطاءها عنوان ذي معنى .
- 5- يجب المحافظة على استمرارية جريان المعلومات من مستوى إلى آخر .
- 6- خلال عملية التصفيه يجب أن تؤخذ في الوقت الواحد فقاعة واحدة .

DFD Components

DFD can represent Source, destination, storage and flow of data using the following set of components -



- **Entities** - Entities are source and destination of information data. Entities are represented by a rectangles with their respective names.
- **Process** - Activities and action taken on the data are represented by Circle or Round-edged rectangles.
- **Data Storage** - There are two variants of data storage - it can either be represented as a rectangle with absence of both smaller sides or as an open-sided rectangle with only one side missing.



- **Data Flow** - Movement of data is shown by pointed arrows. Data movement is shown from the base of arrow as its source towards head of the arrow as destination.

إن الـ DFD ممكن أن يمثل أي مستوى من مستويات تجريد النظام. في الواقع إن الـ DFD نفسه ممكن تجزئته إلى عدة مستويات وهذه المستويات تمثل التعمق للتفاصيل الوظيفية، إن هذه المستويات تبدأ من الصفر فأكثر .

إن الـ level 0 للـ DFD يسمى Fundamental System model أو Context Model يمثل جميع عناصر الـ Software بفاعاة واحدة ومن ثم فإن المستوى واحد سيمثل بتفاصيل أكثر (ممكن أن يحوي على خمس إلى ست ففاعات حيث إن كل ففاعاة بـ level 1 هي دالة جزئية Sub Function من الـ Software الرئيسي .

إن الأشكال الأساسية المستخدمة لرسم الـ DFD هي :-

External Entity

-1

: هو منتج أو مستهلك للمعلومات الداخلة أو الخارجة من الففاعاة الخاصة بالـ Software المعمول له الـ Model .

Process

-2

: هو محول Transformer خاص بالمعلومات (Function) الذي يقع داخل حدود النظام.

Data Flow

-3

: هو Data Flow على شكل مستقيم له رأس سهم يشير إلى اتجاه جريان البيانات.

data store

-4

: هو مخزن للبيانات المخزنة ليتم استخدامها من قبل واحد أو أكثر من الـ Function أو Process وممكن أن تكون بهيئة بسيطة مثل Queue , Stack أو هيئة معقدة مثل قاعدة بيانات علائقية .



مثال على خلق مخطط انسياب البيانات

نظام التسوق عبر الانترنت (online shopping system)

Levels of DFD

- **Level 0** - Highest abstraction level DFD is known as Level 0 DFD, which depicts the entire information system as one diagram concealing all the underlying details. Level 0 DFDs are also known as context level DFDs.

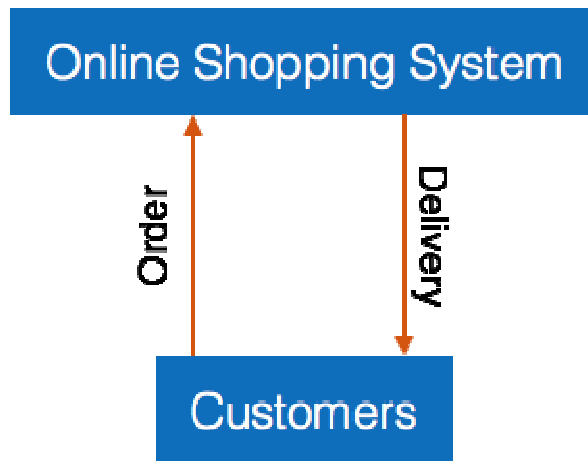


Figure: Level 0 of DFD for Online Shopping System

- **Level 1** - The Level 0 DFD is broken down into more specific, Level 1 DFD. Level 1 DFD depicts basic modules in the system and flow of data among various modules. Level 1 DFD also mentions basic processes and sources of information.

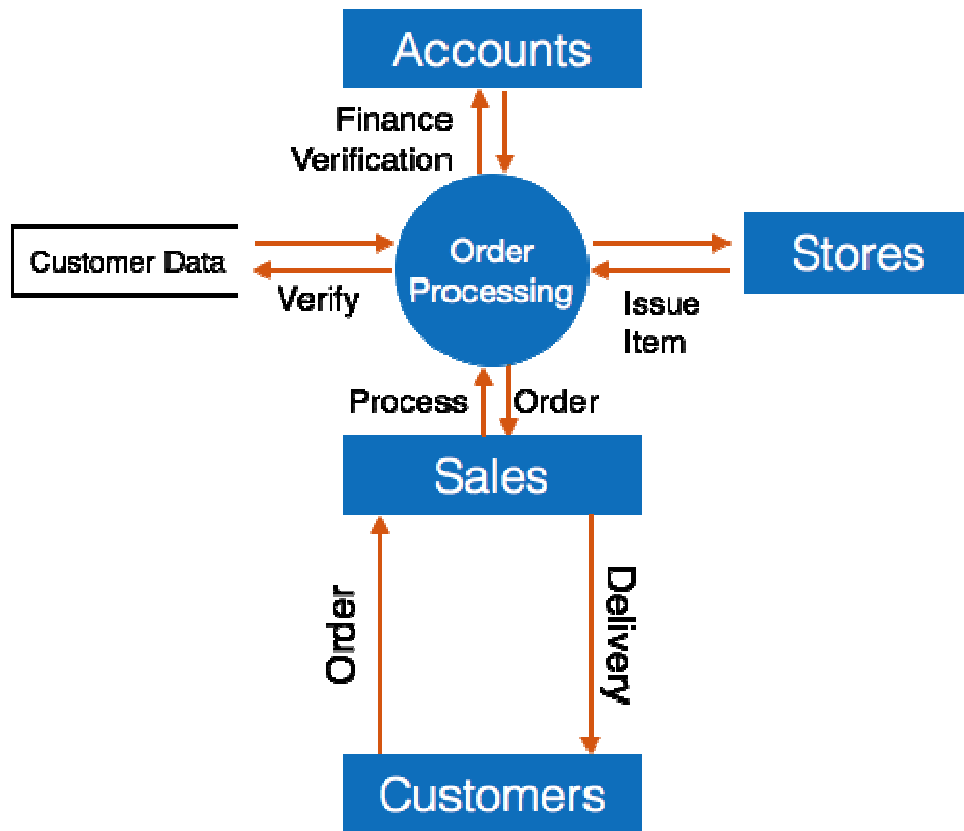


Figure: Level 1 of DFD for Online Shopping System

Level 2 - At this level, DFD shows how data flows inside the modules mentioned in Level 1.

Higher level DFDs can be transformed into more specific lower level DFDs with deeper level of understanding unless the desired level of specification is achieved.