

Chapter three

Software Requirements and Analysis

1.1 Software Requirement

The requirements for a system are the descriptions of the services provided by the system and its operational constraints. These requirements reflect the needs of customers for a system that helps solve some problem such as controlling a device, placing an order or finding information.

المتطلبات لنظام ما هي وصف للخدمات المقدمة من قبل النظام والقيود التشغيلية عليها. تعكس هذه المتطلبات احتياجات العملاء للنظام الذي يساعد في حل بعض المشكلات مثل التحكم بجهاز ما أو العثور على معلومات معينة.

Requirements engineering (RE):- The process of finding out, analyzing, documenting and checking these services and constraints. The goal of requirement engineering is to develop and maintain sophisticated and descriptive 'System Requirements Specification' document.

هي عملية اكتشاف وتحليل وتوثيق ومراجعة هذه الخدمات والقيود. الهدف من Requirements engineering هو تطوير والحفاظ على وثيقة "مواصفات متطلبات النظام" بشكل وصفي .

We need to write requirements at different levels of detail because different types of readers use them in different ways. To distinguish between them we can use the term user requirements to mean the high-level abstract requirements and system requirements to mean the detailed description of what the system should do.

نحتاج إلى كتابة المتطلبات بمستويات مختلفة من التفصيل لأن هنالك أنواع مختلفة من القراء التي تستخدمها بطرق مختلفة. للتمييز بينهما يمكننا استخدام مصطلح "متطلبات المستخدم" التي تعني متطلبات مجردة عالية المستوى ومتطلبات النظام التي تعني الوصف التفصيلي لما يجب أن يفعله النظام.

1.2 Requirements Types

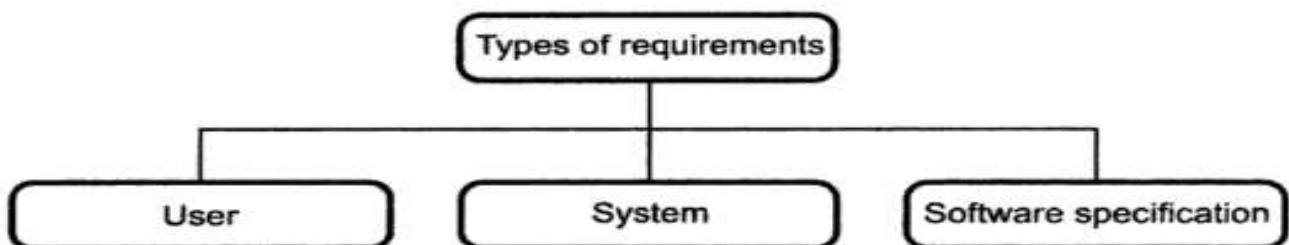


Figure (3-1):- Types of requirements

User requirements:- Are statements, in natural language with simple tables and diagrams (that are easily understood), of what services the system is expected to provide. The user requirements should describe the functional and non-functional requirements in such a way that they are understandable by system users without detailed technical knowledge. It written for customers.

متطلبات المستخدم: - هي عبارات مكتوبة بلغة معينة مع جداول بسيطة ورسوم بيانية (يمكن فهمها بسهولة) ، للخدمات التي من المتوقع أن يوفرها النظام. يجب أن تصف متطلبات المستخدم المتطلبات الوظيفية وغير الوظيفية بطريقة يمكن فهمها من قبل مستخدمي النظام دون معرفة تقنية مفصلة. تكتب هذه المتطلبات للعملاء.

System requirements:- are more detailed descriptions of the software system's functions, services, and operational constraints. It written as a contract between client and contractor (the software developers).

متطلبات النظام: - هي وصف أكثر تفصيلاً لوظائف او الخدمات التي يقدمها النظام والقيود التشغيلية عليها. تكتب كعقد بين العميل والجهة المتعاقد معها (مطوري البرامج).

Software specification:- it is detailed software description that can serve as a basis for design or implementation. Typically it is written for software developers.

مواصفات البرنامج: - هو وصف للبرامجيات بشكل تفصيلي التي يمكن استخدامه كنقطة بداية لمرحلة التصميم أو التنفيذ. عادة ما يتم كتابتها لمطوري البرامجيات.

Functional requirements:- These are statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations. In other words, functional requirements for a system describe what the system should do.

المتطلبات الوظيفية: - هي عبارة عن عبارات للخدمات التي يجب أن يوفرها النظام ، وكيف يجب أن يتفاعل النظام مع مدخلات معينة وكيف يجب أن يتصرف النظام في مواقف معينة. بمعنى آخر ، المتطلبات الوظيفية لنظام ما تصف ما يجب أن يفعله النظام.

Non-functional requirements:- These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and constraints imposed by standards. Non-functional requirements often apply to the system as a whole, rather than individual system features or services.

المتطلبات غير الوظيفية: - هذه قيود على الخدمات أو الوظائف التي يقدمها النظام. وهذه القيود تشمل الوقت ، والقيود المفروضة على عملية التطوير ، والمعايير. غالبًا ما تنطبق المتطلبات غير الوظيفية على النظام ككل ، بدلاً من الخصائص او الخدمات التي يقدمها النظام بشكل فردي.

1.3 The software requirements documents

Also known as software requirements specification (SRS) is the official statement of what the system developers should implement and it include user requirements and system requirements. It should be organized so that both system customers and software developers can use it. The following diagram illustrates possible users of the requirements document and how they use it:

يُعرف أيضًا بمواصفات متطلبات البرمجيات (SRS) وهو بيان رسمي لما يجب على مطوري النظام تنفيذه ويشمل متطلبات المستخدم ومتطلبات النظام. يجب ترتيب بحيث يمكن استخدامها من قبل كل من مستخدمي النظام ومطوري البرمجيات. ويعني تنوع المستخدمين المحتملين أن وثيقة المتطلبات يجب أن تتسم بالمرونة بين توصيل المتطلبات إلى العملاء ، وتحديد المتطلبات بشكل تفصيلي دقيق للمطورين والأشخاص الذين سوف يختبرون النظام ، وتضمين معلومات بالتغييرات المحتمل حدوثها للنظام. والتي يمكن أن تساعد مصممي النظام على تجنب قرارات التصميم المقيدة ومساعدة مهندسي الصيانة الذين يجب عليهم تكييف النظام مع المتطلبات الجديدة.

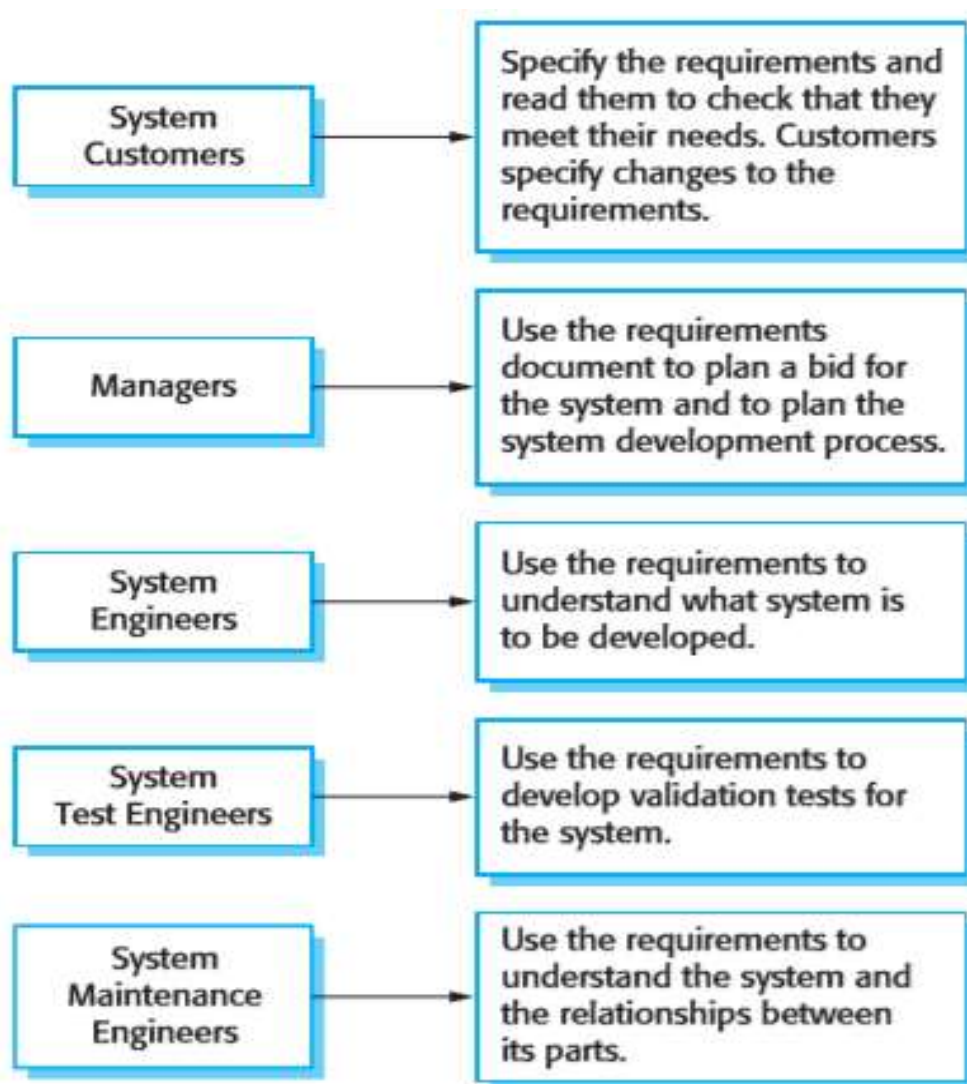


Figure (3-2): Users of requirements document

3.4 Requirements analysis

Requirements analysis is a software engineering task that bridges the gap between system level requirements engineering and software design.

Requirements analysis provides the software designer with a representation of information, function, and behavior that can be translated to data, architectural, interface, and component-level designs.

3.5 Software requirements analysis phases

- (1) Problem recognition.
- (2) Evaluation.
- (3) Modeling.
- (4) Specification.
- (5) Review.

3.6 Analysis principles

1. The **information domain** of a problem must be represented and understood.
2. The functions that the software is to perform must be defined.
3. The behavior of the software must be represented.
4. **The models** that depict information, function, and behavior must be partitioned in a manner that uncovers detail in a layered (or hierarchical) fashion.

-Information Domain

يحوي على كل data objects التي تحوي numbers, text, images, audio or video والعلاقات relationships بين البيانات وهيكلة البيانات information structure

3.7 Analysis Model

The analysis model consists of:

- 1- data model (shows relationships among system objects)
- 2- function model (description of the functions that enable the transformation of system objects)
- 3- behavioral model (manner in which software responds to events from the outside world)

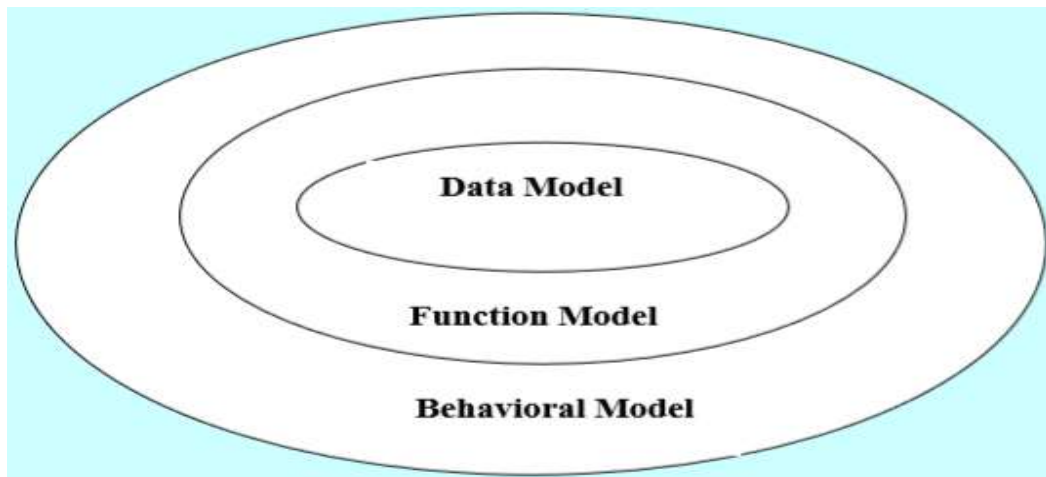


Figure (3-3): Analysis model

1- data model

يجيب عن مجموعة من الاسئلة المحددة التي ترتبط باي نظام معالجة بيانات data processing application

- ١- ماهي primary data objects البيانات الاولية التي سيتم معالجتها من قبل النظام
- ٢- ماهو composition التراكيب لكل نوع من data object وماهي الخصائص التي تصفه
- ٣- ماهي العلاقات relationships بين هذا object وبقية objects
- ٤- ماهي relationships بين objects والمعالجة processes التي ستجري عليها

To answer these questions data modelling methods make use of the entity relationships diagram (ERD), The ERD enables a software engineer to identify data objects and their relationships using a graphical notation.

- Data model component

1- data object

A data object is a representation of almost any composite information (which mean something that has a number of different properties or attributes). Anything have three dimensions (height, width, and depth) could be defined as an object. For example, a person or a car Figure (3-4).

person , هو تمثيل لاي معلومة مركبة composite information (شيء يملك عدد من الخصائص والصفات المختلفة) مثل ,
car

اي شيء يملك 3- dimensions (height , width, depth)

Therefore, the data object can be represented as a table as shown in Figure (3- 5).

2- Attributes

Data 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. Referring to the data object car, a reasonable identifier might be the ID number.

يعرف الخصائص التي تصف data object مثل (الاسم name of data object) او الحالة instance وممكن تعريف احد هذه الصفات key كمفتاح لل data object

3- Relationship

Data object can be connected to one another via relationship.

Relationship can be to one, one to many, many to many

Figure (3-5) For example:

- A person owns a car.
- A person is insured to drive a car.

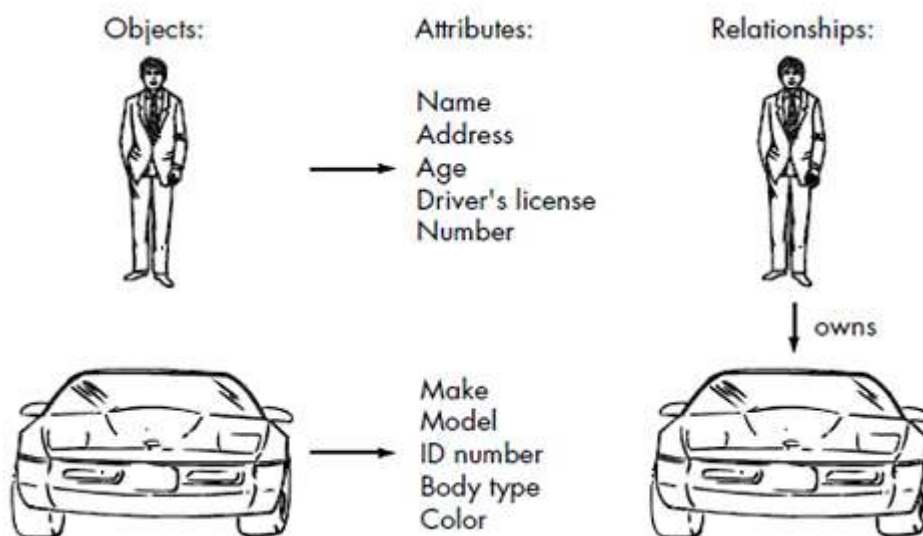


Figure (3-4): data object

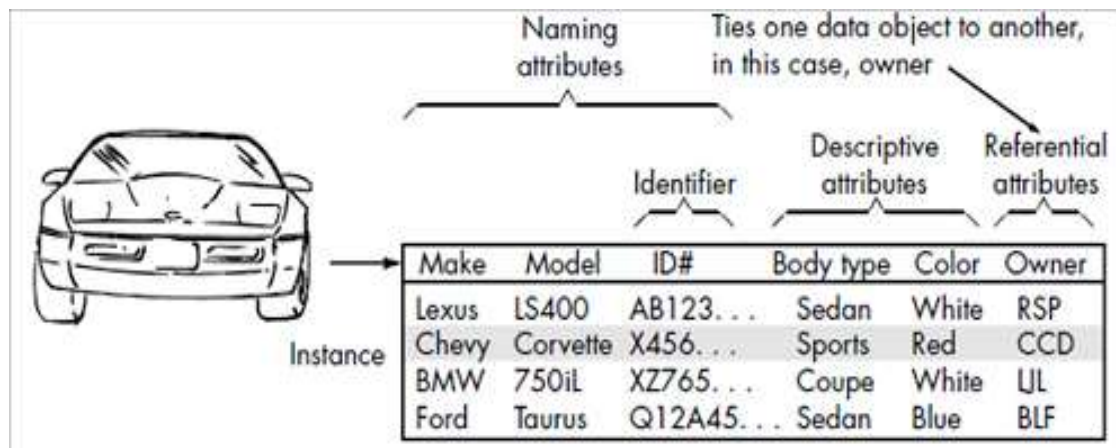


Figure (3-5): data object table

2- function model

- 1- identify functions that transform data object
- 2- identify how data flow through the system.
- 3- Represent producers and consumers of data

3- behavioural model

- a- identify different state of system
- b- Specify events that cause the system to change state

3.8 Cardinality and modality:

Cardinality is the maximum number of objects that can participate in a relationship. how many occurrences of object X are related to how many occurrences of object Y.

Cardinality: the data model must be capable of representing the number of **occurrence** objects in a given relationship.

The cardinality of an object- relationship pair expressed as following:-

1- One-to-One (1:1):

An occurrence of [object] 'A' can relate to only one occurrence of [object] 'B,' and an occurrence of 'B' can relate to only one occurrence of 'A.'

Object A ممكن يرتبط مرة واحدة ب object B ونفس الشيء بالنسبة object B

مثال:- person and identification مثل علاقة الانسان بالهوية او

Ex: in a school database, **each** student has only **one** student ID, and **each** student ID is assigned to only **one** person.

1- 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.'

يرتبط مرة واحدة او اكثر ب object B و يرتبط مرة واحدة فقط من Object A

Ex: Mother can have many children and a child have only one mother

1- 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.'

Object A ممكن يرتبط مرة واحدة او اكثر ب object B و object B ممكن يرتبط مرة واحد او اكثر من Object A مثل

Ex: Uncle can have many nephews while a nephew can have many uncles

Modality: provide whether or not a particular data object must **participate** in the relationship or not

Modality of a relationship = 0 if no need for relation or optional relation

Modality of a relationship = 1 if it must need for relation

مثل نظام برمجيات يستخدم في ورشة تصليح السيارات لتحديد فيما اذا كان هنالك عطل في السيارة ام لا لذلك لغرض اجراء التصليحات يجب وجود زبون، وقد تحتاج الى عدة تصليحات او قد يظهر التشخيص انها لا تحتاج الى تصليحات.

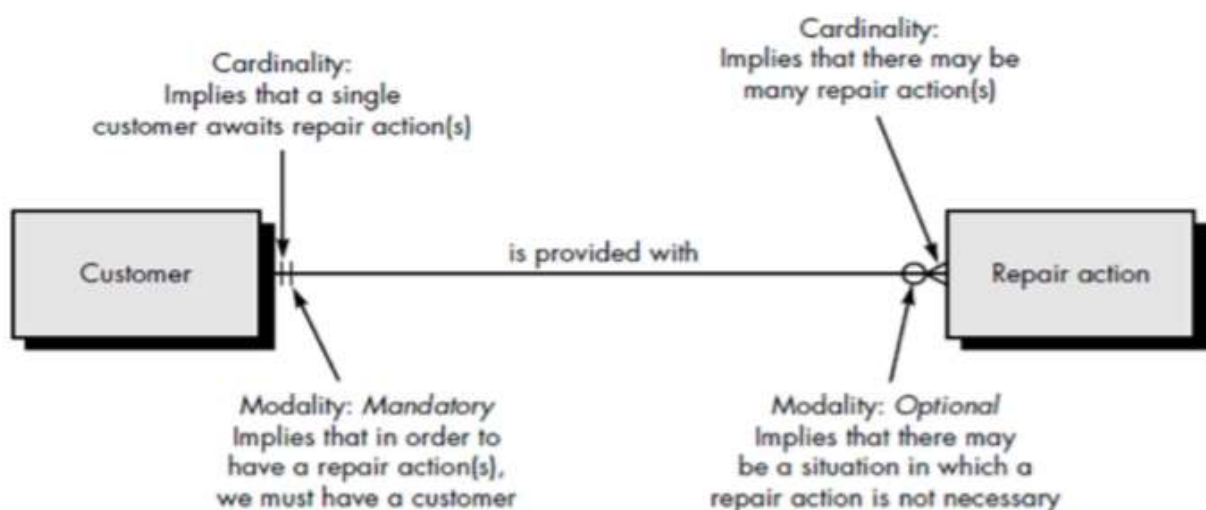


Figure (3.6) Cardinality and Modality

3.9 The analysis modelling approaches

A model is an abstraction of the system being studied (using some kind of graphical notation).

- The analysis model objectives:-

- To describe what the customer requires,
- To establish a basis for the creation of a software design
- To define a set of requirements that can be validated once the software is built.

Analysis modelling approaches:

1- Structure analysis

Is a widely used for analysis model, considers data and the processes that transform the data as separate entities. Data objects are modeled in a way that defines their attributes and relationships. Processes that manipulate data objects are modeled in a manner that shows how they transform data as data objects flow through the system.

يستخدم بكثرة في analysis model ويهتم ب data وعمليات تحويلها الى data model التي تعرف الخصائص attributes والعلاقات relationships لتلك البيانات

2- Object – oriented analysis

This approach to analysis modeling focuses on the definition of classes and the manner in which they collaborate with one another to effect user requirements.

يقوم بالتاكيد على كيفية تعريف class والطريقة التي يتصل class بالآخر لتحقيق user requirements

3.10 The elements of structured analysis model

The structure of analysis model can be expressed in the figure (3-9) **that contain:-**

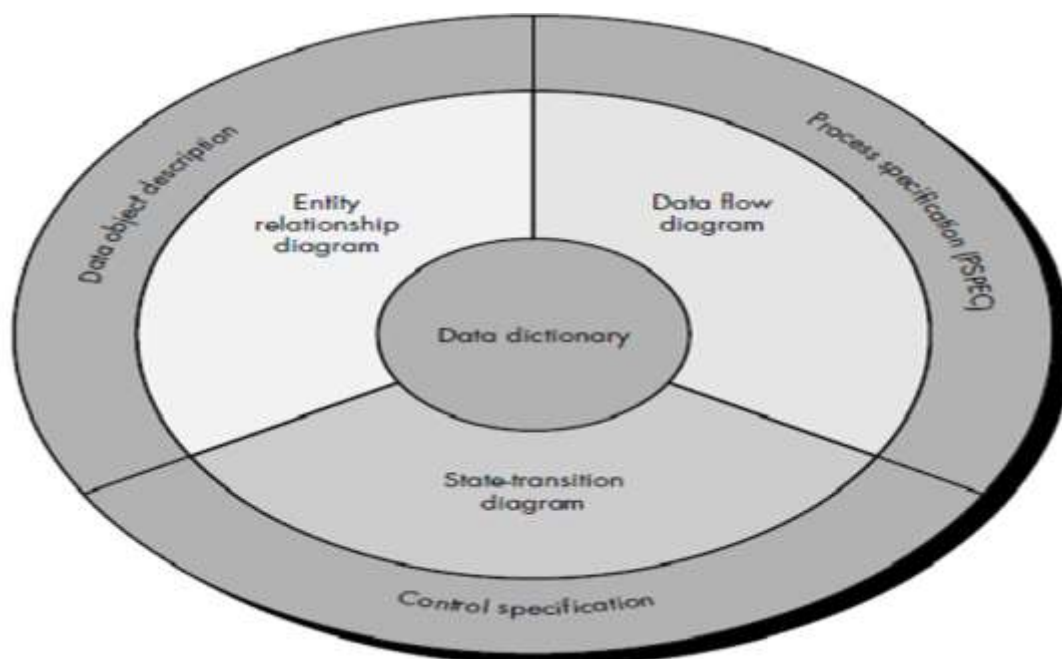


Figure (3-9): The elements of structured analysis model

1- Data dictionary (D.D) : is a repository that contains description of all data objects consumed or produced by the software

هو مستودع يحوي وصف لكل data objects التي يحتاجها او ينتجها النظام وتكون بشكل قائمة منظمة عناصرها data input , output and analyst user and system فهم جيد لل stores

- The data dictionary most contains the following information
 - Name : the primary name of the data or control item, the data store or an external entity.
 - Alias : other name used for the first entry.
 - Where – used / how used: a listing of process that used data or control item and how it is used (e.g., input to the process, output from the process, as a store, as an external entity.
 - Content Description : رمز لتمثيل المحتويات
 - Additional information : information about data types and limitations

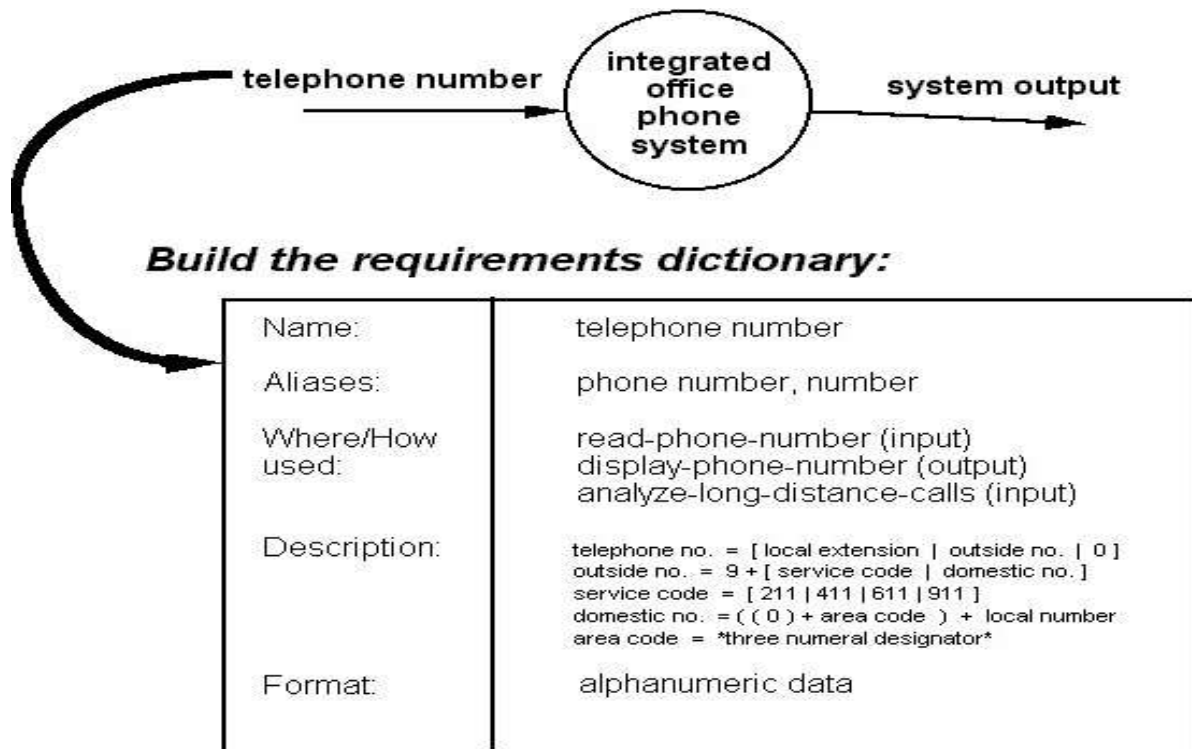


Figure (3-10): data dictionary

2- Entity Relationship Diagram (ERD)

Entity relationships diagram enables software engineer to identify data objects and their relationships using graphical notation. The primary components for ERD are: data objects, attributes, relationships

- **Data objects:** represented by labelled rectangle
- **Relationships:** are indicated with a labelled line connecting objects

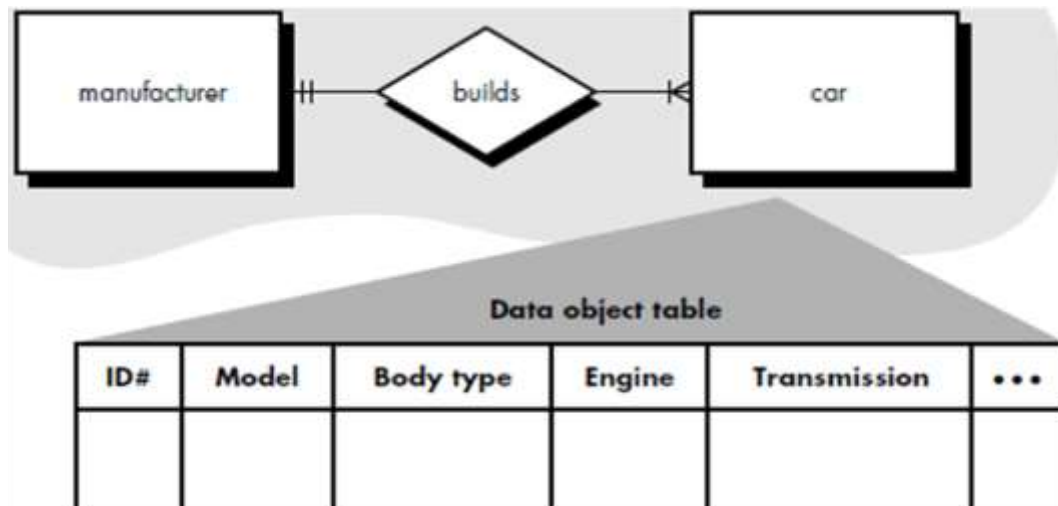


Figure (3-11): Entity Relationship Diagram

3- Data Flow Diagram (DFD)

DFD is a graphical representation that depicts information flow and the transforms that are applied as data move from input to output. A data flow diagram, also known as a data flow graph or a bubble chart. The data flow diagram may be used to represent a system or software at any level of abstraction.

مخطط تدفق البيانات ويسمى أيضا بمخطط الفقاعة ويعرف على أنه عبارة عن رسوم صورية بسيطة تستخدم لتمثيل software system

Input data, processing, output data

Data flow Diagram components

- Data Flow (Arrow)
- External entity (source and destination)
- Process (Circle)
- Data store

4- STD (state transition diagram)

is a type of diagram to describe the behaviour of system through external events. STD require a finite number of states (represents the various modes of behavior) of the system and the manner in which transitions are made from state to state.

5- The process specification (PSPEC)

Is used to describe all flow model processes. The content of PSPEC are text and a program design language (PDL) describe process algorithm, tables, mathematical equations, diagrams or charts

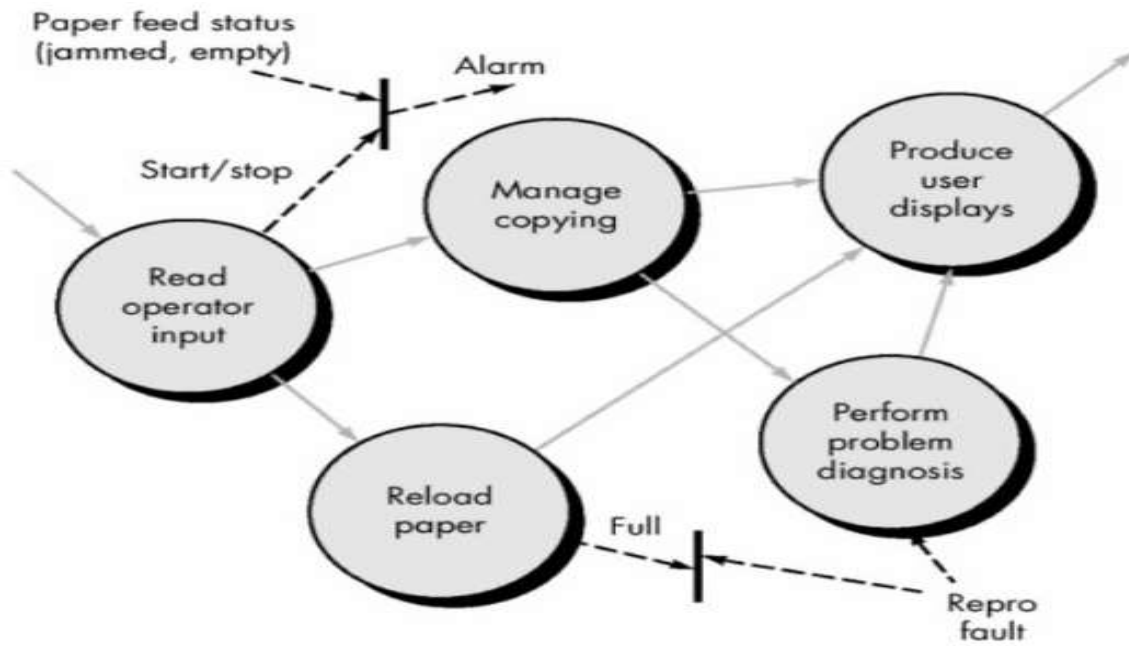


Figure (3-12): state transition diagram

Data flow diagram

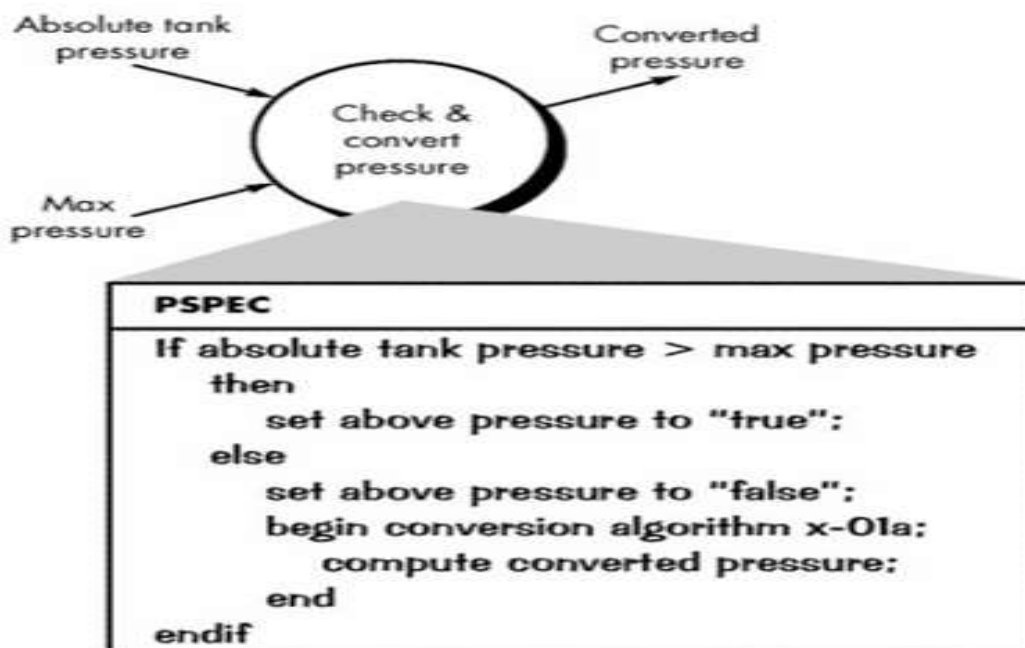


Figure (3-13): The process specification