03rd of February 2023 online lesson

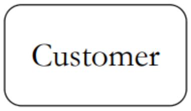
Logical Data Modeling (LDM)

LDM models the systems data processed by the processes identified in DFM. It consists of a diagram called Logical Data Structure (LDS) and associated textual descriptions. LDS illustrates how data interrelates to each other and shows how business rules apply on system's data.

Components of an LDS

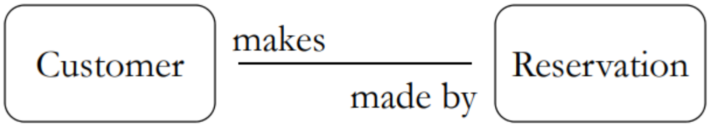
1. Entities

- A collection of logically associated and uniquely identifiable objects or concepts
- Property of an entity is called an attribute
- Name should reflect the type of the entity, not an instance of that entity



2. Relationships

- How entities relate to each other
- Physical relationships (physical links in a data structure)
- Logical relationships (business associations or rules)



3. Degree of a relationship

- Also known as Cardinality of relationship
- Number of occurrences of each entity participating in a given relationship
- Three types (one to one, one to many, many to many)

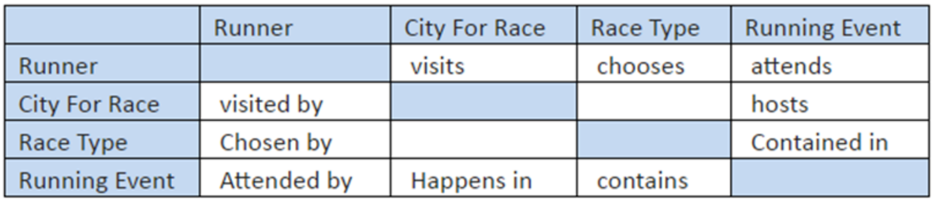
4. Optionality of a relationship

- Shows whether the relationship exists for all occurrences of participating entities

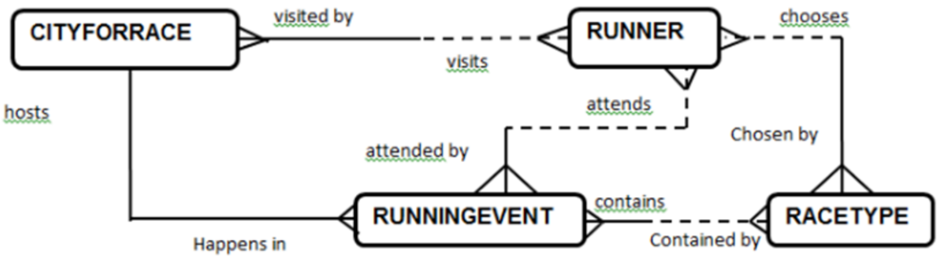
Entities	“something about which an organisation keeps data”	e.g. customer
Attributes	“the properties of an entity”	e.g. address
Relationships	“the connections between entities”	e.g. customer places order

Entity Matrix

- Helps to identify the relationships among entities
- Provides all possible pairings of entities to identify relationships among them in the interested system
- Associations between each pair of entities are checked for once
- Does not provide the full description of a relationship



	Runner	City For Race	Race Type	Running Event
Runner		visits	chooses	attends
City For Race	visited by			hosts
Race Type	Chosen by			Contained in
Running Event	Attended by	Happens in	contains	



Designs the proposed system

Logical Data Flow Modeling (Logical DFM)

Physical DFM shows

- How data is actually processed and
- Where data is actually stored in the current system

Logical DFM shows

- How data should be processed and
- Where the data should be stored in the proposed system

Logical DFM consist of a set of DFDs and associated textual descriptions

LDM of the proposed system

- adds new entities required to support any new functionalities of the proposed system to the LDS
- provides logical data stores (entities) identified in the LDM for logical DFM
- removes any temporary data store that simply halts data temporarily

Logical DFM of the proposed system

- starts from elementary processes
- as location of a process only indicates a physical constraint, removes it from all processes

- as the person who actually did a process in physical DFM now feeds inputs to the corresponding process in the logical DFM, transforms the location into an external entity
- removes any purely human activity and replaces it with an external entity
- adds new processes required to support any new functionalities of the proposed system
- reconstructs the hierarchy by regrouping logical processes based on their functionality
- describes the functionality of elementary processes in pseudo code and write EPD

Physical Design of Database

Maps logical schema to relational schema

Logical Schema	**Relational Schema**
Entity	Table
Attribute	Field
Instance of an entity	Record of a table
Unique attribute	Primary key

- normalizes all the relations in relational schema to 3NF
- tabulates a Table Specification and a Record Specification for each relation in the normalized relational schema
- provides physical design for the construction of the database

Data Dictionary

- is an integral part of database
- holds information about the database and the data that it stores (data about data - metadata)
- contains the actual database descriptions used by the Database Management System (DBMS)

Develops and test the proposed system

Testing

Test Cases

A test case is a set of actions executed to verify a particular feature or functionality of a software application. Test cases are documented by the Quality Assurance team while the software development is going on.

Software testing techniques

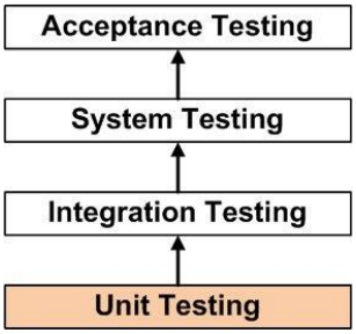
White box testing

- takes internal implementation of a software to derive test cases to test the software
- performed in the early stages of the testing process
- usually applied for testing relatively small program units
- analysis of the program code determines how many test cases are required to exercise all the internal components of the software (statements, branches, paths) adequately

Black box Testing

- software to be tested is treated as a black box and its behavior is examined by studying the inputs and outputs
- knowing the intended functionalities of a software, tests are conducted to see whether the software can deliver them
- test cases are derived from the requirement specification of the software to be tested

Software testing types



In software development,

- software systems are built with sub-systems
- sub-systems are built with individual program units such as functions or classes

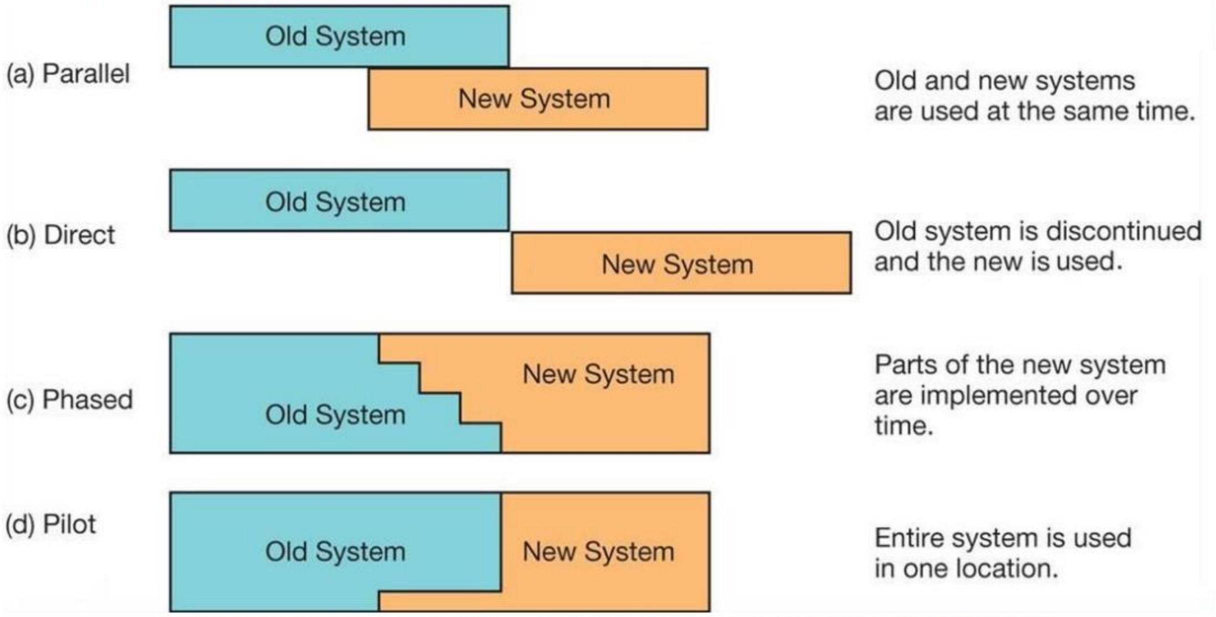
Therefore, software testing,

- starts with testing of these individual program units (Unit Testing, usually carried out by programmers, white box techniques are used)
- continues with the testing of the integration of these units (Integration Testing, usually carried out by integration testers or test teams, can either be white box or black box)
- and the testing of the system's functionality as a whole (System Testing, usually carried out by test teams independent of the programmers who developed the system, black box techniques are used)
- finally ends with testing to see whether the system is acceptable to the users (Acceptance Testing, usually carried out by test teams independent of the programmers and/or users who developed the system, black box techniques are used)

Deploys the developed system

Software deployment

Software deployment includes all the activities that make a software system available for use. The general deployment process consists of several interrelated activities with possible transitions between them. These activities can occur at the developer's side or at the user's side or both.


Old System
(a) Parallel
New System
Old and new systems are used at the same time.
Old System
(b) Direct
New System
Old system is discontinued and the new is used.
New System
(c) Phased
Old System
Parts of the new system are implemented over time.
(d) Pilot
Old System
New System
Entire system is used in one location.

(Source: https://www.studyblue.com/notes/note/n/exam-3-ppt-slides/deck/721189)

Parallel Deployment

Parallel deployment is a method of hardware or software migration that involves using the existing and new systems simultaneously until the implementation is judged to be completed and satisfactory.

During the transition, users work with both systems as they gradually learn the new software. There is generally some duplication of effort as, for example, data must be entered in both systems. That duplication may lead to data quality issues. Parallel adoption is the most foolproof method. However, it can also be the most time-consuming option and is usually the most expensive.

Direct Deployment

This is the simplest methods of deployment. Direct deployment is a hardware or software migration method that involves getting rid of the existing system and transferring all users to the new system simultaneously.

Implementation is faster with direct deployment than other deployment methods. As the old system is no longer available, users cannot put off learning the new system. Furthermore, the complete and simultaneous implementation also avoids issues that can arise when users are working with different software or hardware. This is the least expensive migration method, assuming there are no major problems.

On the negative side, however, it can be hard on users to have to learn the new system immediately. The method also involves a lot of problems that arise during implementation can be unpredictable, plentiful and serious, and fallback systems may be inadequate to deal with them. However, if the two systems are incompatible, direct deployment may be the only viable method.

Pilot Deployment

A pilot deployment is a hardware or software migration method that involves rolling out the new system to a small group of users for testing and evaluation.

During the pilot implementation, the users of the test group can provide valuable feedback on the system to make the eventual rollout to all users go more smoothly. Once the test group has approved the system, it can be rolled out across the organization. The testers can then help train other users for the new system. On the other hand, the test group could also determine whether the system is a viable option for the organization.

Phased Deployment

Phased deployment is a hardware or software migration method that involves incremental implementation of a new system.

A phased migration might be conducted either by implementing the entire new system in some locations or business units or by implementing separate modules of the system independently until the implementation is complete.

Because everything is not rolled out at once, the organization doesn’t have to deal with all the potential implementation issues at the same time. Furthermore, information learned from early implementation stages can be applied to guide the rest of the process, so that there are fewer issues as the implementation continues. A phased rollout also allows users to adjust to the new system gradually.

On the other hand, it can be confusing to have groups of users working with different systems or to have users working with elements of different systems. That confusion can also lead to data quality issues.

Maintenance

The maintenance phase of the SDLC occurs after the software is in full operation. The maintenance phase involves making changes to hardware, software, and documentation to support its operational effectiveness. It includes making changes to improve a system's performance, correct errors, deal with security issues, or address new user requirements. To ensure that the modifications do not disrupt operations or degrade a system's performance or security, organizations use change management standards and procedures.

Describes system implementation with commercial-off-the-shelf packaged system

Advantages of using Commercial Off-The-Shelf (COTS) Packages

COTS software systems are designed for both horizontal and vertical markets. In order to cater for the needs of many, vendors spend an enormous amount of time and effort to incorporate world-class functionality into their systems. Duplicating such expense to develop similar functionality in custom developed software systems, in many cases, is prohibitively expensive.

COTS software systems tend to be far more configurable than custom developed software systems. In other words, they can accommodate change by adjusting parameters to a far greater degree than is usual for custom developed software systems. Moreover, package environments often come with capabilities that users do not to use immediately but are there for future exploitation as and when required.

A COTS software system can be implemented in significantly less time than a custom software system development and often at substantially less cost.

Disadvantages of using COTS Packages

COTS software systems can be highly complex and usually include many features that will be never used. Given the software tends to be large and complicated, it may take a long time to learn how to use the system properly. They may also require to alter the way that the work is done in order to fit in with the way that the system has been designed.

If the organization's business processes are unique, then the COTS software systems may need to be customized or the business processes may need to be changed to accommodate them.

As the competitors can also buy and use the same COTS software system, it is very difficult to gain any competitive advantage from using it.

Advantages of using custom developed software systems

Custom developed software systems are designed specifically to the requirements and built to operate exactly as needed. Moreover, they can be modified as the business changes or altered to further improve their performance. Custom developed software systems can also be integrated with the existing systems and any future systems as required.

The custom developed software systems can provide functions that are needed and none that that are not needed. They also operate in the way that users are used to work which means that they will be easier to use.

Being able to perform tasks that the competitors cannot do means that organizations can gain real competitive advantage with custom developed software solutions. Given that custom developed software can be listed as an asset of an organization, it can add value to the business.

Disadvantages of using custom developed software systems

Custom developed software systems require a large initial investment and the development process can take a long time

Business Process

A set of activities, responsibilities, resources and data flows that interact to accomplish a business function.

Business process modeling

Business process modeling involves an in-depth analysis and then an optimization of the business process by removing inefficiencies and bottlenecks.

Business process gap analysis

As no COTS software solution has been specifically designed to meet any organization's unique requirements, there will be a gap between the business processes supported by the existing systems and those supported by the COTS software system. It is imperative that to understand this gap well before the implementation begins and ensure that the organization can accept this gap without degrading its business performance.

Business process mapping

Business process mapping is a group activity performed by teams of subject matter experts that gather to draw step-by-step diagrams to document how business is carried out. This invaluable tool is mostly used by consultants and business professionals to capture the current state of business operations in preparation for business improvement initiatives.

Business process mapping can also be very beneficial in helping to increase productivity among staff, implementing or decommissioning systems, streamlining processes, and protecting knowledge capital.

In most COTS software systems, there will be gaps between what the organization needs and what the system delivers. For each such gap, organizations have to decide whether to ignore it (remove the requirement and just live with the tool), change how they do something outside the solution (modify the business process) or build something to bridge the gap (extend the solution).

If the COTS software system is extended, it is required to fully specify the requirements for those new capabilities just as done for any new product development effort.

Reference – Teaches’ guide 2017