

Software Engineering

Lecture 6

System Modeling – Part I

Dr. Obuhuma James

Description

System modeling as a crucial element of system development. The topic will be covered in two lectures. This topic aims at exploring part I of the process by looking at the first set of system models required for the implementation phase. These are context diagrams, Data Flow Diagrams (DFD), Entity Relationship Diagrams (ERD), among others. In each case, the need for the model is discussed, standard symbols used to draw the model are outlined, and finally an example of the model is outlined.

Learning Outcomes

By the end of this topic, you will be able to:

- Describe the need for system models in the system development process.
- Draw context diagrams and transform them into different levels of Data Flow Diagrams based on system requirements.
- Develop Entity Relationship Diagrams based on system requirements.

Overview of System Modeling

System modeling is the process of developing abstract models of a system. Each model aims at presenting a different view or perspective of the system. System models are developed based on the requirements generated through the requirements engineering process. Thus, the requirements document discussed in the previous topic forms the basis for the development of system models. In most cases, system modeling is anchored over the Unified Modeling Language (UML) which is an ISO standard that outlines a set of models required to offer graphical representation of systems.

System modeling helps in communicating the requirements in a graphical manner. This helps analysts to understand the functionality of the system as well as facilitate communication with customers. According to Sommerville [1]:

- Models of the existing system are used during requirements engineering to clarify what the existing system does. They can be used as a basis for discussing its strengths and weaknesses which in return lead to generation of requirements for the new system.

- On the other hand, models of the new system are used during requirements engineering to explain the proposed requirements to other system stakeholders. Engineers use these models to discuss design proposals and to document the system for implementation.

Types of System Models

There are many different types of system models. Each of them serves a different purpose or communicates a specific message. These models include context diagrams, data flow diagrams, use case diagrams, class diagrams, entity relationship diagrams, activity diagrams, communication/collaboration diagrams, sequence diagrams, statechart diagrams, deployment diagrams, among others.

This topic will look at the first set of diagrams, namely, context diagrams, data flow diagrams, and entity relationship diagrams. The next topic will explore more models, particularly those that fall under the UML set of models.

Context Diagram and Data Flow Diagram (DFD)

A context diagram is a type of data flow diagram. It is a structured analysis and design tool that allow the analyst to comprehend the system and subsystems visually as a set of interrelated data flows [2]. According to Kendall and Kendall [2], data flow diagrams graphically categorise data processes and flows in business systems. They provide a clear understanding of the interrelatedness of systems and subsystems. This becomes a better analysis of proposed system and aid in communication of current system knowledge to users. They give freedom from committing to the technical implementation too early in the development process.

There are two main sets of symbols that can be adopted in the drawing of data flow diagrams. There are as specified by Gane & Sarson and Yourdon & De Marco. Figure 1 shows symbols used in Kendall and Kendall [1] which are the ones proposed by Gane and Sarson. This course adopts the symbols.


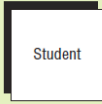

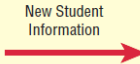

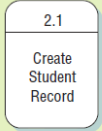
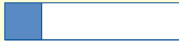
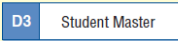
Symbol	Meaning	Example
	Entity	
	Data Flow	
	Process	
	Data Store	

Figure 1. Standard Symbols for Data Flow Diagrams [2]

Yourdon and De Marco instead proposed the use of a circle to represent processes, a rectangle to represent external entities, two horizontal, parallel lines or sometimes an ellipse to represent a data store and finally, curved or straight line with arrowhead indicating flow direction to represent data flow.

- Entities are basically sources or destinations of data beyond the boundaries of a system. They represent departments, businesses, people, or machines, among others. Entity names should be nouns.
- Data flows indicate the flow of information between an entity and process or process and data store. The arrowhead shows the direction of flow of data. Each data flow should have a description against it.
- Processes denote a task or action being performed on data in the system. The naming should relate to the task the process performs.
- Data stores act as depositories for data that gives room for examination, addition, and retrieval of data. They represent databases, files, cabinets among others. D1, D2, D3 and so on are unique reference numbers used to represent data stores.

Figure 2 shows an example of a context diagram for an airline reservation system [1] with one main process called Airline reservation system. The model has three entities with clear flow of data shown and labeled. These are passenger, travel agent and airline. A context diagram

is sometimes referred to as a Context-Level Data-Flow Diagram or a Level-0 Data Flow Diagram. It should have only one process symbol representing the main system. Context diagrams should not have data stores.

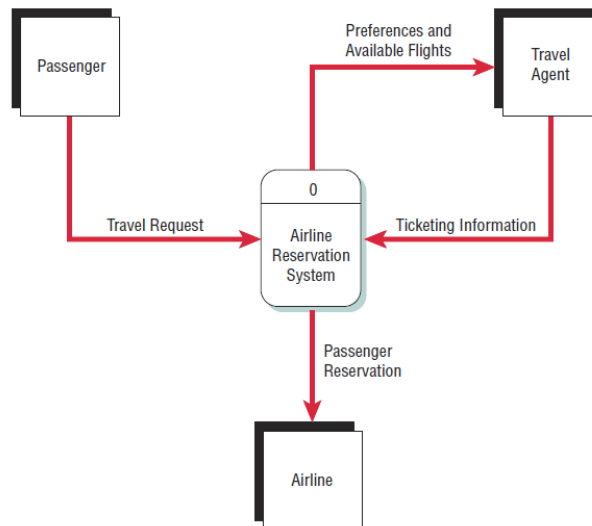


Figure 2. Example of a Context Diagram [2]

Data Flow Diagrams (DFD) are built from a context diagram. They come in different levels depending on how deep they expound on any given process. Hence, processes have to be numbered in levels. Figure 3 shows an example of a data flow diagram.

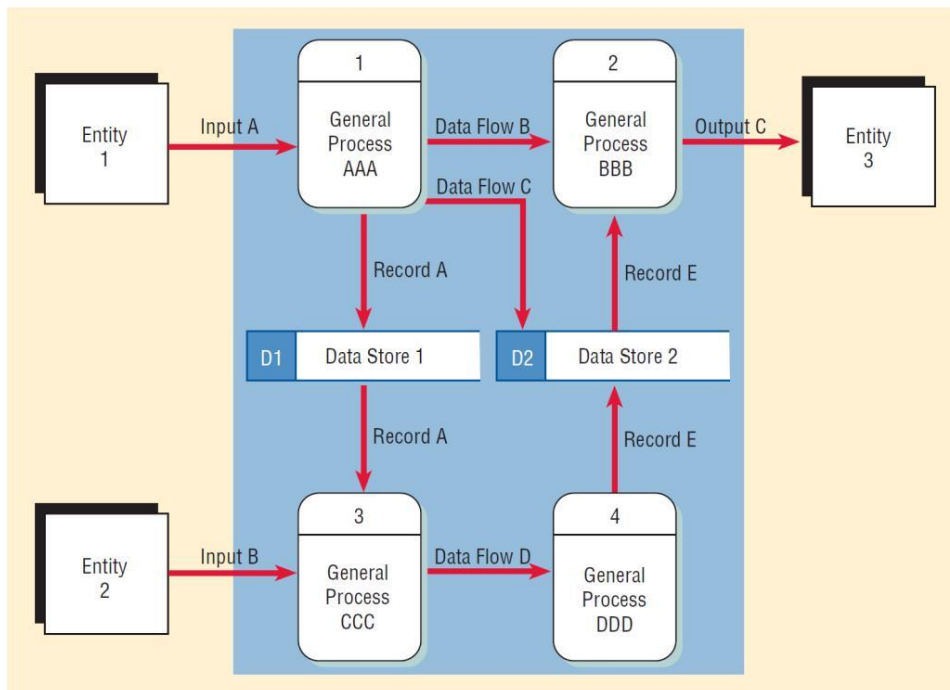


Figure 3. Example of a Data Flow Diagram [2]

Any given data flow diagram has a number of processes with data flowing into and out of each process. Whenever a need for showing more details within a particular process arises, the process in question is decomposed into a number of smaller processes in a lower level DFD. Thus, the starting point at level 0 is a context diagram, whose single process is labeled as 0 signifying "Level-0 DFD". A decomposition of the level-0 process leads to "Level-1 DFD", while the next decomposition leads to "Level-2 DFD", and so on.

Context Diagrams and Data-Flow Diagrams were initially made for systems analysis and design. Over time, they have been elevated to other purposes including capture and communication of interactions and flow of data between business processes.

Entity Relationship Diagram

An Entity Relationship Diagram (ERD) shows the relationship of a set of entities within an organization [2]. They offer another way of showing the scope of a system.

- Entities are real world things representing a person, place, or thing.
- Relationships basically depict the connection between entities. They can be of four different types, namely, one-to-one (1:1), one-to-many (1:N), many-to-one (N:1), and many-to-many (M:N).
- Entities are normally represented together with their associated attributes.

The following process could be followed in the creation of Entity Relationship Diagrams:

1. Identification of entities in the organization
2. Choice of major entities to represent as a way of narrowing the scope of the problem.
3. Determination of attributes for the chosen entities.
4. Confirmation of correctness of the identified entities through data gathering.
5. Establishing relationships between the entities.
6. Drawing the model.

Figure 4 outlines special symbols used in the development of ERDs where:

- Rectangle: Represents Entity sets.
- Ellipses: Attributes
- Diamonds: Relationship Set
- Lines: They link attributes to Entity Sets and Entity sets to Relationship Set

- Double Ellipses: Multivalued Attributes
- Dashed Ellipses: Derived Attributes
- Double Rectangles: Weak Entity Sets
- Double Lines: Total participation of an entity in a relationship set


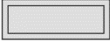





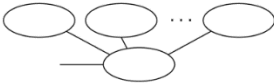




Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute
	Total Participation of E_2 in R
	Cardinality Ratio 1: N for $E_1:E_2$ in R
	Structural Constraint (min, max) on Participation of E in R

Figure 4. ERDs Standard Symbols [3]

Figure 5 shows an example of an ERD for a simple company. According to the example ERD, four entities can be identified, namely, employee, project, department, and dependent. Entities are normally named using nouns. Each entity is represented with its associated attributes, with the primary attribute underlined. The relationship between the various entities has been shown and labelled using verbs as a way to show an action. Furthermore, the type of relationship is clearly shown using 1's and N's.

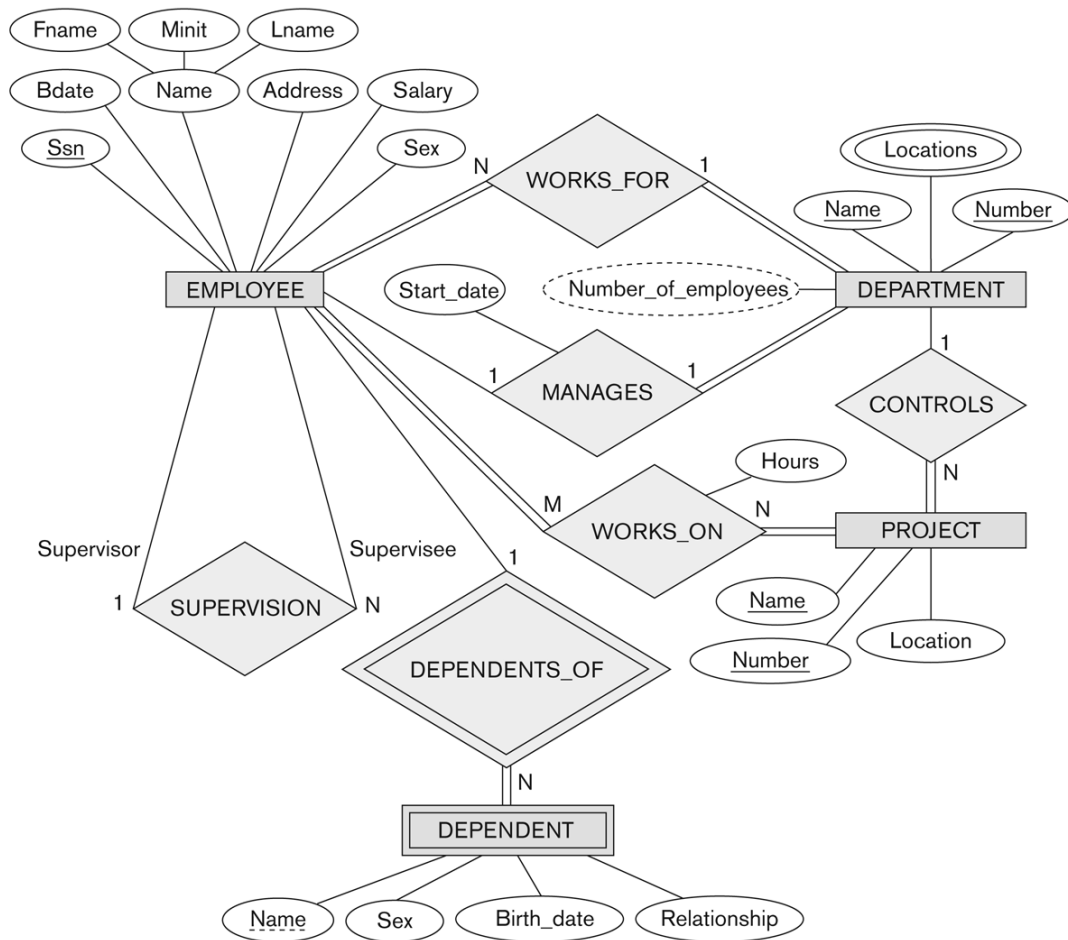


Figure 5. Example of an ERD [3]

According to Elmasri and Navathe [3], the use of ERDs is a popular example for displaying database schemas. Thus, an ERD can be implemented as a database since by design and nature, it is a blueprint of a database. Class diagrams formulated based on UML standards are another way of demonstrating the concepts of entity relationships used in numerous commercial design tools. Class diagrams will be covered in the next topic, which will be covering Part II of System Modeling.

Summary

The topic has established a foundation for system modeling by outlining the reasons behind the use of system models. The various types of system models have been outlined. The first three models, namely, context diagrams, data flow diagrams, and entity relationship diagrams have been discussed extensively. In each case the discussion sought to outline the need for

the model, standard symbols used to draw the model, and finally an example of the model. The next topic will explore other models, majority of which are defined under the UML standard of system models.

Check Points

1. Discuss any three reasons why system modeling is essential.
2. Differentiate between a context diagram and a data flow diagram.
3. Describe the reason behind creation of many levels of data flow diagrams.
4. Discuss the benefit of data flow diagrams in the system development process.
5. Discuss the benefit of entity relationship diagrams in the system development process.
6. Describe the relationship between entity relationship diagrams and the structure of a database.

Learning Resources

Core Textbooks

1. Sommerville, I., Software Engineering, 10th Edition, Addison Wesley, 2016.

Other Resources

2. Kendall, K. E. and Kendall, J. E., Systems Analysis and Design, 8th Edition, Pearson, 2011.
3. Elmasri, R., & Navathe, S. B. (2021). Fundamentals of Database System.

References

- [1] Sommerville, I., Software Engineering, 10th Edition, Addison Wesley, 2016.
- [2] Kendall, K. E. and Kendall, J. E., Systems Analysis and Design, 8th Edition, Pearson, 2011.
- [3] Elmasri, R., & Navathe, S. B. (2021). Fundamentals of Database System.
- [4] Pries, K. H., & Quigley, J. M. (2010). Scrum project management. CRC press.
- [5] Schwaber, K. (2004). Agile project management with Scrum. Microsoft press.
- [6] Gotterbarn, D., Miller, K., & Rogerson, S. (1997). Software engineering code of ethics. Communications of the ACM, 40(11), 110-118.
- [7] Pressman, R. S. (2005). Software engineering: a practitioner's approach. Palgrave macmillan.