Systems Security, Quality, and Control

Data Flow Diagrams

Figure 11W-1 shows a simple data flow diagram for a mail-in university course registration system. The rounded boxes represent processes, which portray the transformation of data. The square box represents an external entity, which is an originator or receiver of information located outside the boundaries of the system being modelled. The open rectangles represent data stores, which are either manual or automated inventories of data. The arrows represent data flows, which show the movement between processes, external entities, and data stores. They always contain packets of data with the name or content of each data flow listed beside the arrow.

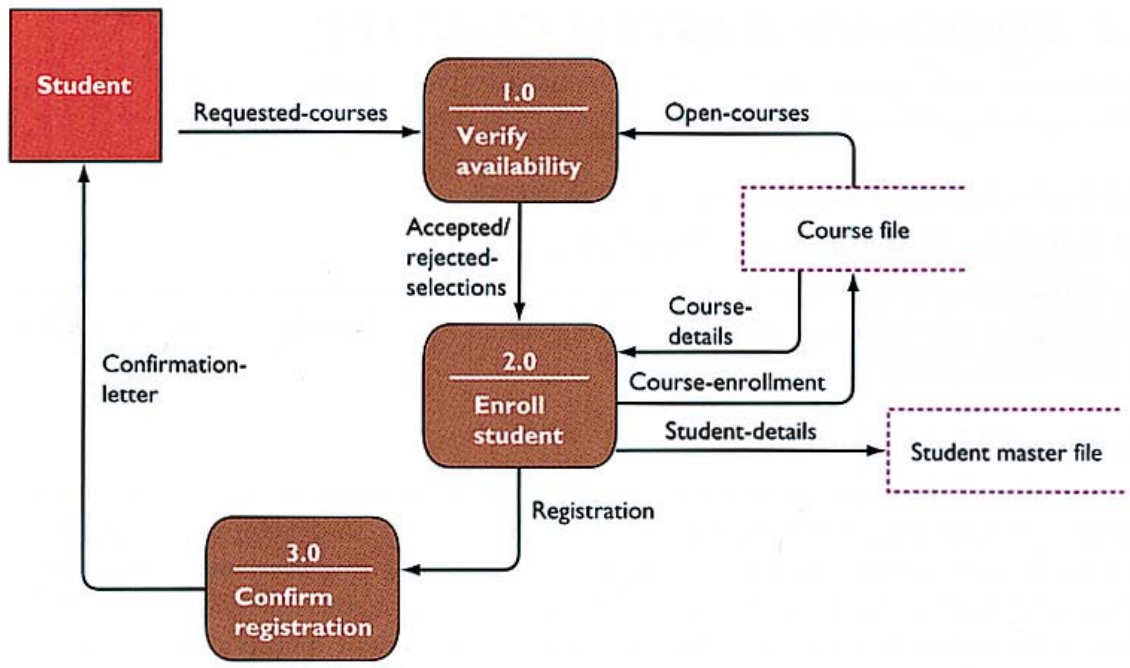

Figure 11W-1 **Data flow diagram for mail-in university registration system.** The system has three processes: Verify availability (1.0), Enroll student (2.0), and Confirm registration (3.0). The name and content of each of the data flows appear adjacent to each arrow. There is one external entity in this system: the student. There are two data stores: the student master file and the course file.

Source: Management Information Systems: Managing the Digital Firm, Eighth Edition by Kenneth C. Laudon and Jane P. Laudon. Page 468.

This data flow diagram shows that students submit registration forms with their name, identification number, and the numbers of the courses they wish to take. In process 1.0, the system verifies that each course selected is still open by referencing the university's course file. The file distinguishes courses that are open from those that have been cancelled or filled. Process 1.0 then determines which of the student's selections can be accepted or rejected. Process 2.0 enrolls the student in the courses for which he or she has been accepted. It updates the university's course file with the student's name and identification number and recalculates the class size. If maximum enrollment has been reached, the course number is flagged as closed. Process 2.0 also updates the university's student master file with information about new students or changes in address. Process 3.0 then sends each student applicant a confirmation-of-registration letter listing the courses for which he or she is registered and noting the course selections that could not be fulfilled.

The diagrams can be used to depict higher-level processes as well as lower-level details. Through levelled data flow diagrams, a complex process can be broken down into successive levels of detail. An entire system can be divided into subsystems with a high-level data flow diagram. Each subsystem, in turn, can be divided into additional subsystems with second-level data flow diagrams, and the lower level subsystems can be broken down again until the lowest level of detail has been reached.

Unified Modelling Language (UML)

Table 11W-1 provides an overview of UML and its components. "Things" are objects and "structural things" allow system developers to describe objects and their relationships. UML uses two principal types of diagrams: structural diagrams and behavioural diagrams.

Structural diagrams are used to describe the relationship between classes. Review Figure 6-9 in Chapter 6, which is an example of one type of structural diagram called a class diagram. It shows classes of employees and the relationships between them. The terminators at the end of the relationship lines in this diagram indicate the nature of the relationship. The relationships depicted in Figure 6-19 are examples of generalization, which is a relationship between a general kind of thing and a more specific kind of thing. This type of relationship is sometimes described as "is a relationship." Generalization relationships are used for modelling class inheritance.

Behavioural diagrams are used to describe interactions in an object-oriented system. Figure 11W-22 illustrates two types of behavioural diagrams: a use case diagram and a sequence diagram. A use case diagram shows the relationship between an actor and a system. The actor (represented in the diagram as a stick man) is an external entity that interacts with the system, and the use case represents a series of

related actions initiated by the actor to accomplish a specific goal. Several interrelated use cases are represented as ovals within a box. Use case modelling is used to specify the functional requirements of a system, focusing on what the system does rather than how it does it. The system's objects and their

Table 11W-1
An Overall View of UML and Its Components: Things, Relationships, and Diagrams

UML Category	UML Elements	Specific UML Details
Things	Structural Things	Classes Interfaces Collaborations Use Cases Active Classes Components Nodes
	Behavioural Things	Interactions State Machines
	Grouping Things	Packages
	Annotational Things	Notes
Relationships	Structural Relationships	Dependencies Aggregations Associations Generalizations
	Behavioural Relationships	Communicates Includes Extends Generalizes
Diagrams	Structural Diagrams	Class Diagrams Object Diagrams Component Diagrams Deployment Diagrams
	Behavioural Diagrams	Use Case Diagrams Sequence Diagrams Collaboration Diagrams Statechart Diagrams Activity Diagrams

Source: Kenneth E. Kendall and Julie E. Kendall, *Systems Analysis and Design,* 5th ed., Upper Saddle River, NJ: Prentice Hall, 2002.

interactions with each other and with the users of the system are derived from the use case model. A sequence diagram describes the interactions among objects during a certain period of time. The vertical axis represents time while the horizontal axis represents the participating objects and actors. Boxes along the top of the diagram represent actors and instances of objects. Lateral bars drop down from each box to the bottom of each diagram, with interactions between objects represented by arrows drawn from bar to bar. The sequence of events is displayed from top to bottom, with the first interaction at the top and the last at the bottom of the diagram. Sequence diagrams are used in system design to derive the interactions, relationships, and operations of the objects in the system.

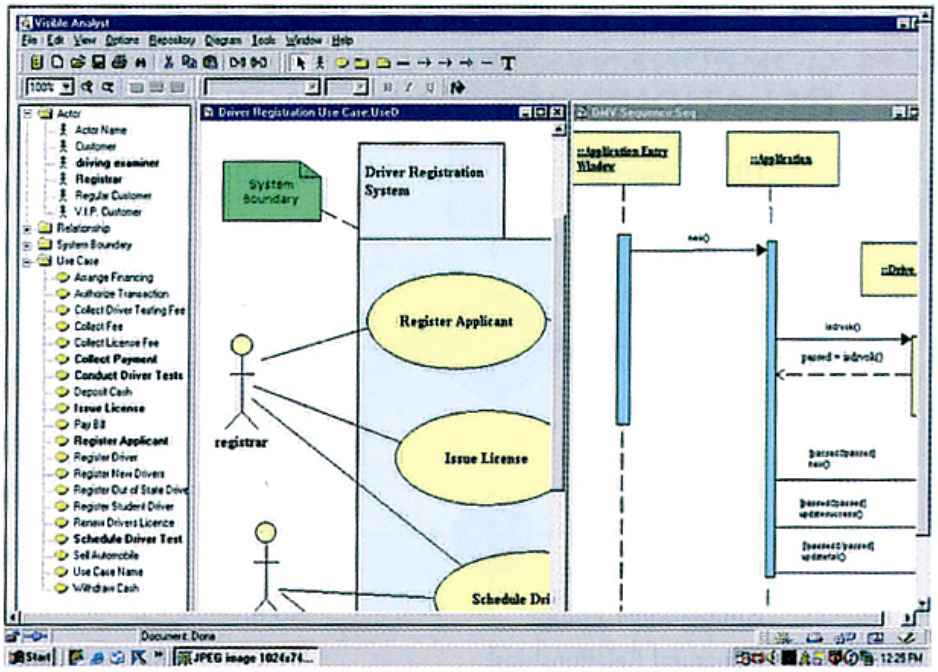

Figure 11W-2 Visible Analyst is a tool for automating object-oriented analysis and design. Illustrated here are examples of use case and sequence diagrams.

Source: Management Information Systems: Managing the Digital Firm, Eighth Edition by Kenneth C. Laudon and Jane P. Laudon. © 2004 Pearson Education Inc., Upper Saddle River, New Jersey, 07458. Page 472.