Chapter 6 – Essentials of Design and the Design Activities

Chapter Overview

There are two major themes in this chapter. The first major theme is about the conceptual foundation principles of systems design. The second is about configuring and designing the environment.

The first three sections in the chapter introduce the concepts of systems design. The first section defines design as distinct from analysis. The objective of systems design is to define, organize, and structure the components of the final solution system that will serve as the blueprint for construction. There are various components that need to be designed, including items such as the application software, the database, the user interface, the network, interfaces to external systems, and internal controls.

The second section compares analysis and design by discussing the different objectives of analysis versus design. It also discusses the different models used for analysis and design. Design models consist of such diagrams as package diagrams, design class diagram, sequence diagrams, communication diagrams, database schema, and user-interface screens.

The final section with the first theme presents the six activities that support Core Process 4, Design the application. These six activities are: design the environment, design application architecture and software, design user interfaces, design system interfaces, design the database, and design system controls and security.

The second major theme in this chapter concerns the detail design of the environment. In other words, the first design activity, design the environment, is covered within this chapter. Various different configurations of the deployment environment are explained, including deploying internal systems, external systems, and remote VPN systems.

Learning Objectives

After reading this chapter, you should be able to:

- Describe the difference between systems analysis and systems design
- Explain each major design activity
- Describe the major hardware and network environment options
- Describe the various hosting services available

Notes on Opening Case

Opening Case

Technology Decisions at Wysotronics, Inc.: This case is a typical situation with many companies today that until recently hosted all of their systems with in-house computers and data centers. With the availability of many different types of hosting services, from colocation to cloud computing, many companies are realizing significant cost savings by using these hosting services. In addition the level
of service and availability has increased because hosting companies have created very robust data centers which provide close to 100% availability. In this case, there are two types of systems – production and supply chain systems that have been moved to a virtual private network, and marketing and sales systems that have been placed in a colocation company that also provides managed services.

**Notes on Running Case:**

**On the Spot Courier Services** (running case): On the Spot is a small, but growing, courier service that needs to track customers, package pickups, package deliveries, and delivery routes. In this chapter the processing requirements for the customers, the home office, and the delivery drivers are reviewed. Customers basically have standard Web page access. The home office has access and update capability to all customer, driver, schedule, and package information. The drivers have real time access and update information about package pickup and delivery. You are asked to determine what equipment is required, using standard off-the-shelf telecommunication and computing equipment. They are also asked to describe any special software required, and to develop a network diagram to support the application.

**The Elements of Design**

**Key Terms**

- **network diagram** – a model that shows how the application is deployed across networks and computers
- **architectural design** – broad design of the overall system structure; also called general design or conceptual design
- **detail design** – low-level design that includes the design of the specific program details

**What is Systems Design?**

The objective of systems analysis is to thoroughly understand the organization’s informational needs or requirements and to document those requirements in a set of specifications. The objective of software construction is to build a system that satisfies those requirements. Systems design, then, is the bridge that takes us from requirements to solution.

**Major Components and Levels of Design**

Today, information systems are deployed on a range of devices—from individual computers and small mobile digital devices to localized networks of computers to large distributed and Internet-connected computers. Each system has unique components that must be designed, so design activities will vary depending on the type of system being developed. Figure 6-1 is a network diagram which illustrates many of the common elements that must be design and developed for a new information system. The major configuration, or structure, of the system must be designed as illustrated in the figure. This high-level design is called architectural design or conceptual design. In addition each of the component parts must also be designed. This lower-level design is often called detail design. Of as shown in the callout boxes in Figure 6-11 the major elements of design consist of the following:
- The environment, including the network and deployment infrastructure. Frequently some portions of the network already exist. Often the network needs to be extended and new elements need to be integrated into existing components.

- The application software, including server based applications and remote applications. Components of the application software may reside on a centralized server, while other components may reside on remote and mobile devices. All components must integrate together for a functioning whole.

- The user interface, which defines the screens and reports on all devices. This can become complex because of the many different devices that may be connected to the system.

- The system interface, which defines communication interfaces with other automated systems. In today's interconnected world, this may become extensive and complex.

- The database, which contains all data structures and deployment methods. Databases are deployed in many different environments, in both single and multiple sites.

- Security and controls, which includes all those considerations of how to protect the system and the data while it resides on any system and in any database. It also includes considerations of how to protect the data while it is in transit from one device to another. Security and controls requires both external security devices as well as built in checks and protections.

Each of components of a final information system will go through both architectural design and then detail design. Figure 6-11 indicates which chapters in the textbook describe the design process for each element to be designed.

**Inputs and Outputs for Systems Design**

In iterative projects analysis and design activities are often done concurrently. However, the first focus of any iteration has to be identifying and specifying the requirements (i.e., analysis); determining the solutions (i.e., design) comes later. Basically, analysis involves decomposition—breaking a complex problem with complicated information requirements into smaller, more understandable components. Analysts then organize, structure, and document the problem domain knowledge by building requirements models. The primary focus of analysis is to “understand.” Design is also a model-building activity. Analysts convert the information gathered during analysis—the requirements models—into models that represent the solution system. The objective of design is to define, organize, and structure the components of the final solution system to have a blueprint for construction of the system. Figure 6-2 shows the transition from analysis to design.

One of the major benefits of object-oriented development is that the design models, in many instances, are simply an extension of the analysis models. Hence the transition from analysis to design is smooth, and in fact analysis and design are often done concurrently.

The primary design models include the following:

- Package diagrams – an architectural design for software components
- Nodes and locations diagrams – an architectural design for the network
- Design class diagrams – programming classes and database design
- Sequence diagrams – detail design of application software
Design Activities

As design decisions are made, especially at the detail level, they are derived from and documented by the building of models. Systems design involves specifying in detail how a system will work when using a particular technology. Each component of the final solution is heavily influenced by the design of all the other components. Thus, systems design activities are usually done in parallel. Each of the activities develops a specific portion of the final set of design documents. Just as a set of building blueprints has several different documents, a systems design package consists of several sets of documents that specify the entire system.

Design the Environment

Every software application must execute in some technology environment. This environment includes the computers and other hardware required for the deployment of the application as well as such things as server computers, desktop computers, mobile computers, firewalls, routers and cabling, fiber optics, and wireless access points. Today’s computing environment has become a world of connected technologies, many of which operate on different protocols and aren’t entirely compatible. A big part of designing the environment is identifying and defining all the types of computing devices that will be required. That includes identifying all the locations and communication protocols necessary to integrate computing hardware.

Design the Application Architecture and Software

In designing the application architecture, the authors include decisions about the structure and configuration of the new system as well as the design of the computer software itself. One of the first steps in this design process is partitioning the software into subsystems. Designing the application architecture is usually a top-down process, with the overall structure defined first and then the detailed design of the various components. Processing requirements influence the technology architecture and the application architecture. Such things as should users be able to access the new system only at work on their desktops or should they also be able to work from home via an Internet connection? Is it necessary to allow remote wireless devices to connect to the system?

Design the User Interfaces

Analysts should remember that to the user of a system, the user interface is the system. It is more than just the screens. It is everything the user comes into contact with while using the system—conceptually, perceptually, and physically. Thus, the user interface isn’t just an add-on to the system. As information systems become increasingly interactive and accessible, the user interface is becoming a larger and more important part of the total system.

Designing the user interface can be thought of as an analysis and a design activity. It has elements of analysis in that the developers must understand the user’s needs and how the user carries out his or her
job. User-interface design is also a design activity in that it requires creativity and conformity to rigorous technology requirements.

**Design the System Interfaces**
A new information system will affect and utilize many other information systems. Sometimes, one system provides information that is later used by another system, and sometimes, systems exchange information continuously as they run. The component that enables systems to share information is the system interface, and each system interface needs to be designed in detail. The form of these interfaces will vary dramatically from a file transfer to a real-time data exchange, to a function call via an application program interface.

**Design the Database**
An integral part of every computer information system is the information itself, with its underlying database. Analysts must consider many important technical issues when designing the database. Many of the technical (as opposed to functional) requirements defined during systems analysis concern database performance needs (such as response times).

It is also not uncommon to have multiple databases, with distinct database management systems. These databases may be distributed across multiple database servers and may even be located at completely different sites. These highly technical issues often require specialized skills from experts at database design, security, performance, and physical configuration.

**Design the Security and System Controls**
The final design activity is ensuring that the system has adequate safeguards to protect organizational assets—the safeguards referred to as system controls. User-interface controls limit access to the system to authorized users. System-interface controls ensure that other systems cause no harm to this system. Application controls ensure that transactions are recorded precisely. Database controls ensure that data is protected from unauthorized access and from accidental loss due to software or hardware failure. Finally network controls ensure that communication through networks is protected.

**Design the Environment**

**Key Terms**

- **local area network (LAN)** – a computer network in which the cabling and hardware are confined to a single location
- **client-server architecture** – a computer network configuration with user’s computers and central computers that provide common services
- **client computers** – the computers at which the users work to perform their computational tasks
- **server computer** – the central computer that provides services (such as database access) to the client computers over a network
- **Hypertext Markup Language (HTML)** – the predominant language for constructing Web
pages and which consists of tags and rules about how to display pages

- **Transmission Control Protocol/Internet Protocol (TCP/IP)** – the foundation protocol of the Internet; used to provide reliable delivery of messages between networked computers
- **three-layer architecture** – a client/server architecture that divides an application into view layer, business logic layer, and data layer
- **view layer** – the part of the three-layer architecture that contains the user interface
- **business logic layer or domain layer** – the part of a three-layer architecture that contains the programs that implement the business rules and processes
- **data layer** – the part of a three-layer architecture that interacts with the data
- **Hypertext Transfer Protocol Secure (HTTPS)** – an encrypted form of information transfer on the Internet that combines HTTP and TLS
- **Transport Layer Security (TLS)** – an advanced version of Secure Sockets Layer (SSL) protocol used to transmit information over the Internet securely
- **content delivery network (CDN)** – a set of server computers, separate from the hosting computers, used to deliver such static content as images or videos
- **hosting** – the process of providing physical servers at a secure location and selling those services to other businesses that wish to deploy Web sites
- **colocation** – a hosting service with a secure location but in which the computers are usually owned by the client businesses
- **virtual server** – a method to partition the services of a physical Web server so it appears as multiple, independent Internet servers
- **cloud computing** – an extension of virtual servers in which the resources available include computing, storage, and Internet access and appear to have unlimited availability
- **Service Level Agreement (SLA)** – part of the contract between a business and a hosting company that guarantees a specific level of system availability
- **Virtual Private Networks (VPNs)** – a closed network with security and closed access built on top of a public network, such as the Internet
- **peer-to-peer connection** – when independent computers communicate and share resources without the need of a centralized server computer

The first activity in the list of design activities is designing the environment. This activity is also listed first because it permeates all the other design decisions. There is an incredible variation in the software systems being deployed today as well as an explosion in the types of devices and configurations that have software applications. In this section, the authors address issues related to three major industry trends in software deployment: software systems deployed entirely within an organization, software systems built for purely external use (in our case, deployed on the World Wide Web via the Internet), and software systems deployed remotely in a distributed fashion (for internal and external use).
Design for Internal Deployment

There are two types of internally deployed software systems: stand-alone systems and internal network systems.

**Stand Alone Software Systems:** A stand-alone system is any software system that executes on a single computing device without connecting externally via an Internet or network connection. Design issues for stand-alone systems are usually straightforward. These systems usually read and write data into files without database access. The biggest issue with stand-alone systems is that they often need to be deployed on various pieces of equipment.

**Internal Network-Based Systems:** An internal network-based system is one that is for the exclusive use of the organization that builds it or buys it. It isn’t meant to be used by anyone except company employees who are located within the organization’s physical facilities. Usually these systems are deployed as part of a local area network (LAN) using client computers and server computers. There are two kinds of systems that can be deployed in a client-server architecture:

- Desktop application systems
- Browser-based application systems

A desktop application is one that uses its own custom designed screens and reports that are part of the application software itself. The advantage of this type of system is that the presentation (i.e., the user interface) and the functionality can be customized to the exact requirements of the users. Examples of these types of systems include graphical or engineering systems in which the processing and presentation requirements are very strict and very intensive.

The other type of internal network system is one that is browser based. In a browser-based system, the presentation of screens and reports to the user’s computers (i.e., the clients) is handled by an Internet browser, such as Internet Explorer, Firefox, Chrome, or Safari. In this configuration, most of the processing and heavy calculation is done by the server and then passed to the client computers as Hypertext Markup Language (HTML) pages.

**Three-Layer Client-Server Architecture:** One effective method of software design is to separate the user-interface routines from the business logic routines and separate the business logic routines from the database access routines. This method of designing the application software is called **three-layer architecture**, consisting of the view layer, the business logic layer, and the database layer. A major benefit of using three-layer architecture is its inherent flexibility. Interactions among the layers are always requests or responses, which make the layers relatively independent of one another. It doesn’t matter where other layers are implemented or on what type of computer or operating system they execute. Multiple layers can execute on the same computer or each layer can operate on a separate computer. Complex layers can be split across two or more computers.

Design for External Deployment

The largest and most rapidly growing arena for new software applications is the deployment of systems that are purely for external use on the Internet. Tremendous growth has occurred in a broad range of online business opportunities including purely online businesses, “brick and mortar” businesses that have extended to the Web, and many home-based and other small businesses. Important issues related to the environment for externally deployed systems include:
Configuration for Internet deployment

Hosting alternatives for Internet deployment

Diversity of client devices with Internet deployment

**Configuration for Internet Deployment:** Configuration of the application for the Internet must address several issues. Almost all Internet-deployed applications use a three-layer architecture. The back end (i.e., the application server and the database server) provides the same functionality as an internally deployed client-server system. The view layer architecture has some similarities but also has more complex requirements due to the varied and insecure nature of the Internet. The view layer consists of the HTML pages that are rendered by a browser.

Implementing an application via the Web has a number of advantages over traditional client/server applications, including such things as accessibility, low-cost communication, and widely accepted implementation standards. Of course, there are problematic aspects of application delivery via the Internet and Web technologies. Issues such as security, throughput, and changing standards are potential problems that must be addressed.

Security is especially important due to the high level of hacking and intrusion. This applies both to data while it resides on the server and while it is in transit. Transportation security is enhance through such standards as Hypertext Transfer Protocol Secure (HTTPS) and Transport Layer Security (TLS). Throughput of high transaction volume is enhanced by server farms where servers are linked together with load-balancing hardware and software. Content delivery networks (CDN) are also a popular technique to enhance overall throughput.

**Hosting Alternatives for Internet Deployment:** Hosting the application software is also an important consideration for Web-based systems. Before deciding on hosting method or a hosting company the primary organization must consider such things as reliability, data center security, support staff, growth capabilities, and physical facilities with connectivity and power protection. Inasmuch as there are many companies now that do provide excellent hosting services, there has been a big migration from home-grown data centers to hosting with service providers.

Hosting companies provide their services in a variety of packages and offerings. A few of the more popular ones include:

- **Colocation** – providing a secure physical facility with stable connectivity where companies can locate their own servers
- **Managed Services** – providing additional hardware and software services such as managing Internet servers, database servers, maintaining the operating systems, providing backup services.
- **Virtual Servers** – providing partial use of a server computer or multiple expanded servers in a virtual configuration
- **Cloud Computing** – an extension of virtual servers where a client can purchase only as much computing capability as needed at the moment and without having to worry about the server configuration. It is an attempt to treat computing capability like a utility such as electricity.

In all of the above, the hosting companies guarantee a certain level of performance through a Service
**Level Agreement (SLA)**. It is not uncommon in today's world for hosting companies to guarantee 100% uptime.

**Diversity of Client Devices with Internet Deployment**: The final issue, that of the diversity of client devices that can access the Internet, must be considered in designing any externally deployed system. This issue will continue to become more important as additional devices become Web enabled. One way to categorize client devices is by size: full-sized computers, mid-sized tablet computers, and small mobile computing devices. Most new Internet software applications have at least two separate view layers to accommodate the disparity in these devices.

**Design for Remote, Distributed Environment**

A remote, distributed environment has characteristics of the internal environment and the external, Web-based environment. As with internal configuration, the software applications for a remote, distributed environment are often internal systems used by employees of a business. As with the external, Web-based deployment, the employees aren’t constrained to a single location; in fact, they can range throughout the world.

**Remote Deployment via Virtual Private Network**: Today, almost all these systems are built by using the Internet and are called Virtual Private Networks (VPNs). A VPN is a network built on top of a public network such as the Internet, which offers security and controlled access for a private group. This type of configuration is built with a secure “pipe” between the remote computers and the home office server, which uses Internet protocols but with more security and control. In order to implement this type of VPN, special software is used to establish a secure connection and to encrypt all data transmissions.

A variation of this configuration can occur when two remote computers need to communicate directly with each other. They can use the home office server to facilitate the connection between them, but after they are connected, they can establish a peer-to-peer connection that continues the communication without any other assistance.

**Diversity of Client Devices**: Software applications deployed remotely have even more complex rendering requirements. Often, specialized equipment is needed to deploy an application to remote employees—for example, when a courier service transmits customer signatures back to the home office upon a package’s delivery.

**RMO Corporate Technology Architecture**

RMO’s main offices consist of the corporate headquarters as well as a large retail store, a manufacturing plant, and a large distribution warehouse in Park City, Utah. Locations of operation centers, warehouses, and data centers are shown in Figure 6-17.