Overview

Unit 9 is organized in four sections, the first being this overview and the introduction. The second section is concerned with strategies used in design — modern structured design, information engineering, rapid application development, joint application development, prototyping and object-oriented design. The third section covers the activities associated with each of the three design phases — configuration, procurement, and design and integration. The last section includes a further installment of the case study on the Orient-Pacific Insurance Corporation, which enables you to practise some of the techniques covered in Unit 9. This final section also includes a unit summary, list of references, glossary of terms and solutions for the self-tests and activity.
Introduction

A logical model defines system data and processes independently of technology. Systems design (also called physical design) is concerned with the evaluation of alternative solutions and the specifications of a detailed computer-based solution. In other words, we use the requirements specified in systems analysis to define hardware, software, communications and interface solutions. Stated differently, systems design is concerned with the four building blocks of information systems (IS) — data, processes, interface and geography — from the designers’ point of view. Several different design options, which are proposed to management, are fully described and evaluated for feasibility.

Your first task is to create a number of technical options. You use a logical model to create options for physical implementation. You have to accommodate all the data and processes defined in this model in your design. You may need to add other physical functions.
Objectives

By the end of Unit 9, you should be able to:

1. Define information systems (IS) design.

2. Describe a number of important IS design strategies, including modern structured design, information engineering, rapid application development, joint application development and object-oriented design.

3. Describe the activities involved in each of the three phases associated with IS design: configuration, procurement, and design and integration.
Strategies for systems design

Similarly to systems analysis, there are different strategies that can be used in systems design. Even though these strategies are often regarded as competing with one another, it is not unusual to find a combination of them being used in practice.

Modern structured design

Structured design is based on a hierarchical, top-down partitioning of the system. On completion, a system comprises a number of programs. Each program comprises a number of modules. Each module comprises a number of lines of program code. A system may comprise many thousands of lines of code. This code is produced or written by programmers. In software design, you specify what the program code must do — that is, you communicate to programmers what they must write. We use a structure chart to show the hierarchy that ties modules to their subsystems. It should be noted that with the emergence of some other design techniques, structured design has lost some of its allure. However, it is still a popular strategy in the mainframe environment.

Structured design (also called top-down design) focuses primarily on the processes in the IS building blocks (discussed in Unit 1). In doing so, we try to design hierarchical modules that have the following two properties:

- they are cohesive
  In other words, the modules should perform a single function only. Cohesive modules are easier to write and maintain. It is also easier to reuse cohesive modules in other systems.

- they are loosely coupled
  In other words, modules should be designed in such a way that their dependence on one another is minimal. It should be clear that the writing and maintenance of loosely coupled modules are easier than modules that have great dependency on other modules. You should ensure that a module is independent of other modules by reducing inter-module communication to a minimum. Coupling is largely a programming issue (rather than a design issue). However, a designer must ensure that all modules are as logically independent of each other as possible.

Engineering approaches have produced further solid, practical principles of what makes ‘good’ module design. These principles can be applied to software and can assist in making software design decisions:

1. distinguish between manager (control) modules and worker (detail) modules

Large bureaucratic organizations have layers of managers. As you ascend this bureaucracy, you have increasing power and ‘span of control’, but decreasing hands-on contact with day-to-day business. In the same way, a modular program design has high-level modules
responsible for program control and low-level modules responsible for
doi ng work (for example, computations, input, output and data
manipulation). In between are ‘middle manager’ modules that do some
control and some work.

2 assign module names carefully

Names are critical to communication (to other programmers and
designers). Names must be meaningful and reflect what a module
actually does.

3 partition modules into a manageable size

If a module is very long, you sub-divide it into smaller, yet logical,
modules. If a module is very short, you integrate it into its parent
module.

4 avoid ‘hard coded’ variables

Modules are functional and should have no data of their own. They
should have no percentages, rates, report headings, and so on, embedded
within them. This data is likely to alter, and changing it is difficult (and
dangerous) if it is embedded in code. You should record all such data in
parameters. These are stored on files so that when data changes, a
parameter file can be changed rather code.

**Information engineering**

Unlike structured design, which is process-centred, information engineering
strategy is data-centred. Using this strategy, we prioritize the applications
that need to be developed; and then use a combination of other design
strategies for the actual development of the applications.

**Prototyping**

As discussed in *Unit 4*, a prototype is a small-scale, representative, working
model of a system. Prototyping is an iterative development strategy which
translates the requirements into a working system that is continually revised.
Prototyping allows for the identification of design problems early on in the
development stages by allowing the users to take an active role in the
development processes. In spite of its advantages, prototyping has several
disadvantages. In particular, in the absence of proper specifications,
prototyping may lead to premature or sloppy design. Therefore, sufficient
attention should be paid to the issues involved in the survey and study
phases.

**Joint application development (JAD)**

As you saw in *Unit 4*, joint application design emphasizes participative
development among the system’s stakeholders. During systems design
sessions, the systems designer acts as a facilitator among the stakeholders
who meet to specify or review systems design issues and deliverables.
Rapid application development (RAD)

The data and process models of an application are first built using structured techniques. A prototype is then used to verify systems requirements and refine data and process models. Through an iterative process of defining and refining models and prototypes, the requirements are then evaluated and used in designing the system.

Object-oriented design (OOD)

Object-oriented design has grown out of object-oriented programming. It brings functional and data design approaches together. An object is a data structure plus all of the processing modules that apply to that data structure. For example, a POLICY data structure for the Orient-Pacific Insurance Corporation would have modules to add a new policy, delete a policy, amend policy attributes, find policy by name of client, and so on. Transactions, or programs, send messages to objects they wish to manipulate.

Reading

‘Systems design approaches’, pp. 395–401 in your text. This reading corresponds to Objective 2 of this unit.

Self-test 9.1

1 Differentiate between systems design and systems analysis.

2 Modules should be highly cohesive and loosely coupled. Why?

3 List three advantages and disadvantages of prototyping.

4 Explain the relationship between prototyping, JAD and RAD.
Activities involved in systems design phases

Similarly to Unit 4, in this unit we will follow the decision analysis and systems design phases in the FAST methodology to show the activities involved and the deliverables produced.

The decision analysis

The purpose of this phase is to identify several alternative solutions and recommend a target solution. The activities involved in this phase are:

• defining candidate solutions
  
  We need to identify candidate solutions using the business requirements outlined in the definition phase of systems analysis. Because of the technical details involved, it is useful to use a matrix (similar to the one shown on page 379 of your text) to summarize the attributes of different options.

• analysing the feasibility of alternative solutions
  
  As discussed in Unit 8, each design option needs to be assessed for operational, technical, economic and schedule feasibility. If you have problems remembering these feasibility tests, please refer back to Unit 8 for their descriptions. The deliverable of this activity is a feasibility matrix (also discussed in the previous unit).

• recommending a system solution
  
  We need to select the most appropriate design solution using cost-benefit analysis and other techniques discussed in Unit 8. The outcome of this activity is a system proposal, which includes a description of the project scope, IS requirements analysis, alternative solutions and feasibility analysis, and a set of recommendations.

Self-test 9.2

1 What are the objectives and end-products of the decision analysis phase?

2 How might an existing technology architecture impact the decision analysis phase?
This phase may be concerned with buy/build decisions for the technology needed to support the requirements of the three focuses of the IS framework. Do you remember these three focuses (we discussed them in Unit 1)? There are six activities involved in this phase:

- **researching technical criteria and options**
  We need to determine the hardware and software requirements generated from the configuration phase by specifying the important criteria, such as functionalities, features and performance parameters, for the proposed system. The deliverables of this activity are a list of potential vendors, product options and technical criteria.

- **soliciting proposals**
  Having researched the technical criteria, we need to solicit proposals from different vendors. This is done through a *request for proposals (RFP)* or a *request for quotations (RFQ)*. RFPs are used to specify the requirements and desired features of the proposed solution, and to indicate their degrees of importance. While some of the requirements and features are mandatory, others may be classified as important but non-critical. In situations where we have already decided on a product, we use RFQs to get information on prices, maintenance and services arrangements, etc.

- **validating vendor claims and performance**
  The proposals of different vendors need to be evaluated in order to ensure that their claims are valid. Proposals which make unsubstantiated claims or do not meet the mandatory requirements are eliminated at this stage.

- **evaluating and ranking vendor proposals**
  The next step involves the use of feasibility and cost-benefit analysis techniques to evaluate the validated proposals. Do you remember how to use these techniques? If not, please refer back to *Unit 8*.

- **awarding contracts**
  The analyst makes a recommendation to management as to which vendor should be awarded a contract. The terms of the contract then need to be negotiated with the winning vendor. The losing vendors also need to be informed of the outcome.

- **establishing integration requirements**
  One of the most important tasks this stage of IS development relates to the integration of the new system into the existing systems. A large number of the issues that need to be addressed at this stage are related to the way the new system fits into the existing portfolio of systems.
**Reading**


These readings correspond to Objective 3 of this unit.

---

**Self-test 9.3**

1. Explain the difference between a request for proposals (RFP) and a request for quotations (RFQ).

2. Distinguish between the terms valuation and evaluation as they apply to the selection of computer hardware and software.

---

**Design and integration**

At this phase we need to develop technical design specifications with a view to achieving the following two objectives:

- to design a user-friendly system that meets the business requirements
- to provide clear and complete specifications to programmers.

If you recall, in Unit 4 we analysed the requirements of an information system in terms of data, processes, interfaces and geography. At this stage we need to establish detailed design requirements. More specifically, the issues related to network requirements (to be discussed in Unit 10), processing methods (also to be discussed in Unit 10), data requirements and database design (to be discussed in Unit 11), and interface design (to be discussed in Unit 12) need to be addressed. In general there are six activities involved in this phase:

- analysing and distributing data

  In Unit 6, you learned about data modelling as a way to document and communicate the data requirements of the target system. Because data models developed at the analysis stages are not necessarily amenable to the design of files and databases, we need to undertake data analysis to prepare data models ready for implementation. Some of techniques which can be used to prepare the data, such as normalization, will be discussed in Unit 11. The outcome of this step is the distributed data model, details of which will be covered in Units 10 and 11.
• analysing and distributing processes
  In this stage, we need to examine the process models (Unit 5) to determine the network requirements of the target system. The outcome of this step is the distributed process model, details of which will also be covered in Units 10, 11 and 13.

• designing databases
  Designing database specifications is an extremely important part of systems design. Not only do databases act as the life-line of a new system, they represent shared resources and, as such, their design must account for the federation of the existing systems as well as systems that will be developed in the future. The outcome of this step are the database design specifications, details of which will be covered in Unit 11, along with other aspects of database design.

• designing computer inputs and outputs
  After designing the required databases, the inputs and outputs need to be specified. Data input capture methods need to be decided upon; the format of the output needs to be determined; and internal controls for ensuring integrity of the input and output should be specified.

• designing online user interface
  Unlike systems developed up to late 1980s, most online systems developed today need to account for the user interface. The details of interface design will be discussed in Unit 12.

• presenting and reviewing design
  The final activity involved in systems design relates to the compilation of all the technical specifications produced in previous activities and presenting them in the form of a technical design statement.

---

**Self-test 9.4**

1. What are the objectives and end-products of the design phase?

2. Distinguish between the terms data analysis and event analysis.

---

**Activity 9.1**

1. Your office is considering an automated system for a small business. Prepare a simple RFP statement for a set of hardware requirements.

2. Match the systems design strategy to the definition provided:
   (i) modern structured design
   (ii) information engineering
(iii) prototyping
(iv) JAD
(v) RAD
(vi) OOD

A Techniques used to refine the object requirement definitions identified earlier during analysis and to define design-specific objects.

B Involves conducting a business area requirements analysis from which information system applications are carved out and prioritized.

C A process-oriented technique for breaking up a large program into a hierarchy of modules that results in a computer program which is easier to implement and maintain.

D The act of building a small-scale representation or working model of the users’ requirements to discover or verify those requirements.

E The merger of various structured techniques with prototyping techniques and joint application development techniques to accelerate systems analysts, designers and builders to jointly define and design systems.

Summary

Systems design concerns specifying software at a level of detail that enables a team of programmers to build a system. In other words, systems design is concerned with the detailed technical specifications of system components.

In this unit, you learned about systems design strategies, including modern structured design, information engineering, rapid application development, joint application development, prototyping and object-oriented design. You should note that, generally, a combination of these strategies is used in practice.

Constructing a general software design is a process of packaging a requirements specification in accordance with the chosen systems architecture. This packaging extends the architecture to detail job steps and isolates modules for online processing. A software development environment helps programmers to use software engineering techniques in order to produce ‘good’ software. Detailed module design, used in structured design, uses coupling, cohesion and other criteria (including naming, module size and separation of data from function in programs) to judge ‘good’ module design.

In this unit, we also covered various activities involved in the three design phases — configuration, procurement, and design and integration. Similarly to Unit 4, in this unit we tried to map these activities into the four focuses of the IS building blocks, examining design issues related to data, processes, networks and interfaces.

In the next unit, Unit 10, we will learn about general design, discussing issues related to application architecture in terms of data, processes, interface and networks. The details of these design aspects will be discussed in Units 11 to 13. More specifically, database design will be examined in Unit 11; human-computer interface design in Unit 12; and Internet design in Unit 13.
References

You should be able to meet the learning objectives of this course and successfully complete you assessment on the basis of your study units and your textbook. These references are not prescribed reading; they are provided to enable you to develop your knowledge beyond the requirements of this course.


Glossary

This glossary provides brief definitions of the main technical terms you encountered in Unit 9. The definitions given here are offered in addition to, or in place of, those found in your textbook.

**Cohesion:** Refers to the degree to which a module’s instructions are functionally related.

**Coupling:** The level of dependency that exists between modules.

**Data analysis:** A procedure that prepares a data model for implementation as a non-redundant, flexible and adaptable file/database.

**Logical model:** A model that defines system data and processes independently of technology.

**Normalization:** The way data attributes are grouped to form stable, flexible and adaptive entities.

**Physical design:** Design specification for hardware, software, communications and interfaces.

**Request for proposals (RFP):** Requests for proposals are used to specify the requirements and desired features of the proposed solution from vendors, and to indicate their degrees of importance.

**Request for quotations (RFQ):** Requests for quotations are used to get information on prices, maintenance and services arrangements of a specific product from vendors.

**Structure chart:** Graphic representation that shows the organization of modules in a program. Basic symbols used in structure charts are a named box for each module; lines that indicate control relationships between modules; and an indication of whether control is a sequence, selection or iteration.

**Systems design:** Systems design is concerned with the four building blocks of IS — data, processes, interfaces and geography — from the designers’ point of view.
Answer key for self-tests and activity

Self-test 9.1

1 Systems design is the evaluation of alternative solutions and the specification of a detailed computer-based solution. Systems analysis is the study of a business problem domain to recommend improvements and specify the business requirements for the solution.

Whereas systems analysis primarily focuses on the logical, implementation-independent aspects of a system (the requirements), systems design deals with the physical or implementation-dependent aspects of a system (the system’s technical specifications).

2 Cohesion refers to the degree to which a module’s instructions are functionally related. Coupling refers to the level of dependency that exists between modules.

Modules should be highly cohesive; that means, each module should accomplish one and only one function. This makes the modules reusable in future programs. Modules should be loosely coupled; that is, they should be minimally dependent on one another. This minimizes the effect that future changes in one module will have on other modules.

3 Advantages of prototyping:

- Prototyping encourages and requires active end-user participation. This increases end-user morale and support for the project because the system appears real to the users.

- Prototyping can increase creativity because it allows for quicker user feedback, which can lead to better solutions.

- Prototyping accelerates several phases of the system development life cycle because it consolidates parts of phases that normally occur one after the other.

Disadvantages of prototyping:

- Prototyping can reduce creativity in designs. The very nature of any implementation can prevent analysts and end-users from looking for better solutions.

- Prototyping often leads to premature commitment to a design. In other words, the configuration and procurement phases get shortchanged.

- When prototyping, the scope and complexity of the system can quickly expand beyond original plans. This can easily get out of control.
There is a relationship between prototyping, joint application development (JAD) and rapid application development (RAD) because they are often used in combination with one another.

Prototyping is a technique for quickly building a functional model of information system using rapid application development tools. It also encourages and requires active end-user participation.

JAD is a technique that complements other systems analysis and design techniques by emphasizing participative development among system owners, users, designers and builders. JAD can be used to speed up the prototyping process.

RAD is actually the merger of various structured techniques with prototyping techniques and joint application development techniques to accelerate systems development.

**Self-test 9.2**

1. The objectives of the decision analysis phase are to:
   - identify candidate solutions
   - analyse those candidate solutions
   - recommend a target system that will be designed and implemented.

   The end-product of the configuration phase is a system proposal intended for the system managers or a steering committee who will make the final decision.

2. The purpose of the decision analysis phase is to:
   - identify candidate solutions
   - analyse those candidate solutions
   - recommend a target system that will be designed and implemented.

With an existing technology architecture, system managers may tend not to look for new technologies due to reasons such as costs and user resistance, thus limiting the technical choices. This makes the identification and analysis of candidate solutions much more difficult.

**Self-test 9.3**

1. A request for proposals (RFP) is used when several different vendors and/or products are candidates and you want to solicit competitive proposals and quotes. A request for quotations (RFQ) is used when you have already decided on the specific product, but that product can be acquired from several distributors.

Therefore, the main difference between an RFP and an RFQ is that the later communicates something more solid and specific to the vendors. An RFP communicates only requirements and desired features to the
prospective vendors, who will then decide on the quotations, while an RFQ solicits something that is more specific, such as configurations, prices, maintenance agreements and servicing.

2 Validation is the verification of claims that a vendor has made for the hardware and software they have proposed. Evaluation is the comparison of vendor proposals against one another to select a winner. Ideally, proposals should be validated before they are evaluated. Otherwise a vendor may be ranked higher on the basis of an incorrect claim.

**Self-test 9.4**

1 The purpose of the design phase is to generate detailed and technical design specifications for a new information system or for modifications and enhancements to an existing system. These design specifications will be passed on to the system builders (developers or programmers) for development and implementation. The output of this phase is a technical design specification (or statement) which will guide the system builders as the project moves on to construction.

2 Data analysis is a procedure that prepares a data model for implementation as a non-redundant, flexible and adaptable file and database. Event analysis is a technique which studies the entities of a fully normalized data model to identify business events and conditions that cause data to be created, deleted, or modified.

**Activity 9.1**

1 Hardware requirements:

- Suitability: The computer configuration must fit in with the user’s requirements (e.g., direct access facilities, hard-copy output in given quantities).

- CPU size: The power of the CPU must be sufficient for current and foreseeable requirements. This should be considered as a mandated requirement.

- Back-up storage size and file access speed: Hard disks should offer sufficient storage capacity and fast file access.

- Reliability: There should be a low expected ‘break-down rate’. There should be back-up facilities and, in the case of a microcomputer, this may mean being able to resort temporarily back to a manual system when the computer breaks down.

- Simplicity: Simple systems are probably best for small organizations.

- Ease of ‘communication’ between the hardware and the user: The system (hardware and software) should be able to communicate well with the user. Software is referred to as ‘user-friendly’ or ‘user-unfriendly’. Similar considerations apply to hardware (e.g., not all
terminals are of standard screen size; the number and accessibility of terminals might also have a bearing on how well the user is able to put data into the computer or extract information.)

- **Flexibility:** The hardware should be able to meet new requirements as they emerge.

- **Security:** Keeping out unauthorized users is easier with more powerful systems, although security can be a major problem for any computer system.

- **Cost:** The cost of the system must be justified by the expected use and benefits to be obtained from it.

- **Hardware and operating system standards:** The industry ‘standard’ for microcomputer hardware and operating systems should be enforced.

- **Network compatibility:** The hardware could be networked through various connection standards. Open system features, such as TCP/IP, are a mandated requirement.

- **Documentation:** User documentation should be clear, to help with internal training of staff.

- **Maintenance:** Will the supplier provide a maintenance service; and, if so, what speed of response will the supplier provide to call-outs? (Most microcomputer hardware comes with a one-year warranty, but what about maintenance after the first year?)

- **Printer speed:** The printer being purchased should be able to print output at the required speed and to the desired quality standard.

2 (i) C
(ii) B
(iii) D
(iv) F
(v) E
(vi) A
Unit 9

Information systems design and RAD
**Course Team (Original course)**

Project manager:  Steve Elliot, OUHK

Course developers:  University of Technology, Sydney
Craig McDonald, Consultant
David Wilson, Consultant

Course designer:  Shannon Timmers, OUHK

**Course Team (Revised course)**

Course developers:  Ali F. Farhoomand, Consultant
Lester W. Yee, Consultant

Course designer:  Linda Chung, OUHK

Course coordinator:  John Mak, OUHK

Member:  Nigel Leung, OUHK

**External course assessor**

Professor NV Balasubramanian, City University of Hong Kong

**Production**

ETPU Publishing Team

**Copyright acknowledgements**

The following copyright materials are reproduced under the terms of a licence issued by Hong Kong Reprographic Licensing Society. No further copying is permitted.

Information in Activity 9.1.

**Acknowledgement**

The OUHK gratefully acknowledges a grant of $100,000,000 from the Hong Kong Government designated for course development.

*B329 Systems Analysis and Design* was developed in part using resources provided by this generous grant.


All rights reserved.
No part of this material may be reproduced in any form by any means without permission in writing from the President, The Open University of Hong Kong.

The Open University of Hong Kong
30 Good Shepherd Street
Ho Man Tin, Kowloon
Hong Kong
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>1</td>
</tr>
<tr>
<td>Introduction</td>
<td>2</td>
</tr>
<tr>
<td>Objectives</td>
<td>3</td>
</tr>
<tr>
<td><strong>Strategies for systems design</strong></td>
<td>4</td>
</tr>
<tr>
<td>Modern structured design</td>
<td>4</td>
</tr>
<tr>
<td>Information engineering</td>
<td>5</td>
</tr>
<tr>
<td>Prototyping</td>
<td>5</td>
</tr>
<tr>
<td>Joint application development (JAD)</td>
<td>5</td>
</tr>
<tr>
<td>Rapid application development (RAD)</td>
<td>6</td>
</tr>
<tr>
<td>Object-oriented design (OOD)</td>
<td>6</td>
</tr>
<tr>
<td><strong>Activities involved in systems design phases</strong></td>
<td>7</td>
</tr>
<tr>
<td>The decision analysis</td>
<td>7</td>
</tr>
<tr>
<td>Design and integration</td>
<td>9</td>
</tr>
<tr>
<td><strong>Summary</strong></td>
<td>12</td>
</tr>
<tr>
<td><strong>References</strong></td>
<td>13</td>
</tr>
<tr>
<td><strong>Glossary</strong></td>
<td>14</td>
</tr>
<tr>
<td><strong>Answer key for self-tests and activity</strong></td>
<td>15</td>
</tr>
</tbody>
</table>