Systems Analysis and Design

Program Design

Chapter 10

Key Definitions

Program design - creating instructions for the programmers

The top-down, modular approach - begin with the "big picture" and gradually add detail

Program design document - all structure charts and specifications needed by programmers to implement the system

التعاريف الرئيسية
تصميم البرنامج - إنشاء تعليمات للمبرمجين
النهج من أعلى إلى أسفل، وحدات - تبدأ مع "صورة كبيرة" وإضافة تدريجيا التفاصيل
وثيقة تصميم البرنامج - جميع المخططات الهيكلية والمواصفات المطلوبة من قبل المبرمجين لتنفيذ النظام

MOVING FROM LOGICAL TO PHYSICAL PROCESS MODELS

Analysis phase - focus on logical processes and data flows

Design phase - create physical process models showing "how" the final system will work

Physical process models convey the "system view" of the new system

الانتقال من نموذجي إلى نماذج العملية الفيزيائية
مرحلة التحليل - التركيز على العمليات المنطقية وتدفقات البيانات
مرحلة التصميم - إنشاء نماذج العملية الفعلية تظهر "كيف" النظام النهائي سوف تعمل
نماذج العملية المادية تنقل "عرض النظام" للنظام الجديد

The Physical Data Flow Diagram

The physical DFD contains the same components as the logical DFD, and the same rules apply (e.g. balancing, decomposition, etc.)

There are five steps to perform to make the transition to the physical DFD

مخطط تدفق البيانات المادية
يحتوي DFD الفعلي على المكونات نفسها مثل DFD المنطقي، وتنطبق نفس القواعد (مثل الموازنة والتحلل وما إلى ذلك)
هناك خمس خطوات لأداء لجعل الانتقال إلى DFD الفعلي

Steps to Create the Physical Data Flow Diagram

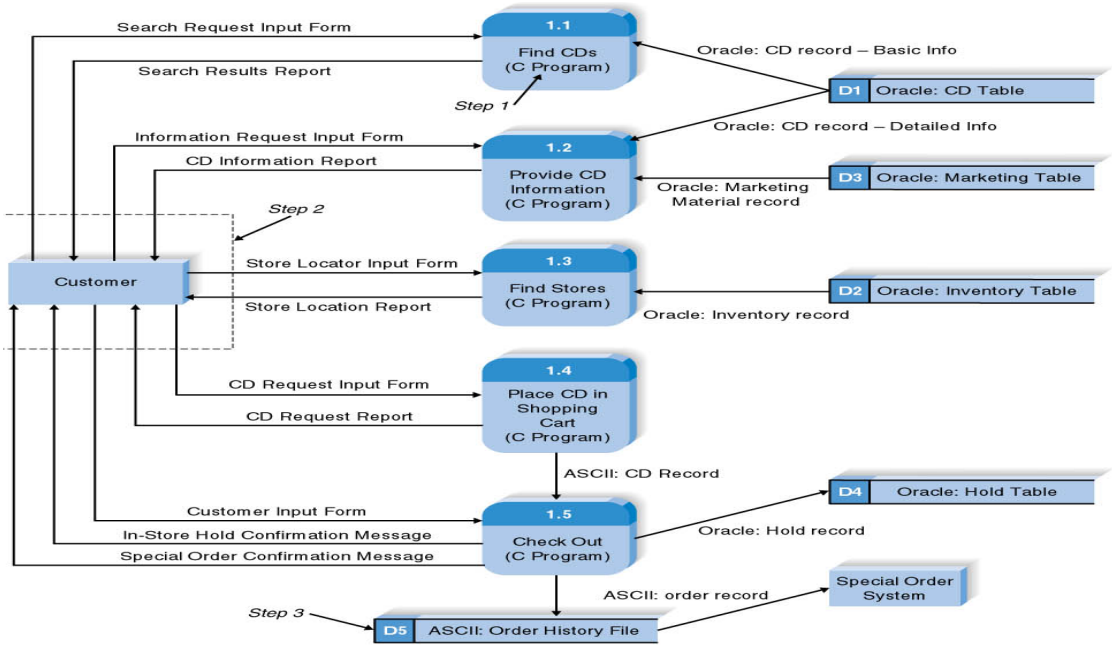
خطوات إنشاء مخطط تدفق البيانات المادية

Step	Explanation
Add implementation references	Using the existing logical DFD, place the way in which the data stores, data flows, and processes will be implemented in parentheses below each component.
Draw a human–machine boundary	Draw a line to separate the automated parts of the system from the manual parts.
Add system-related data stores, data flows, and processes	Add system-related data stores, data flows, and processes to the model (components that have little to do with the business process).
Update the data elements in the data flows	Update the data flows to include system-related data elements.
Update the metadata in the CASE repository	Update the metadata in the CASE repository to include physical characteristics.

CASE = computer-aided software engineering; DFD = data flow diagram.

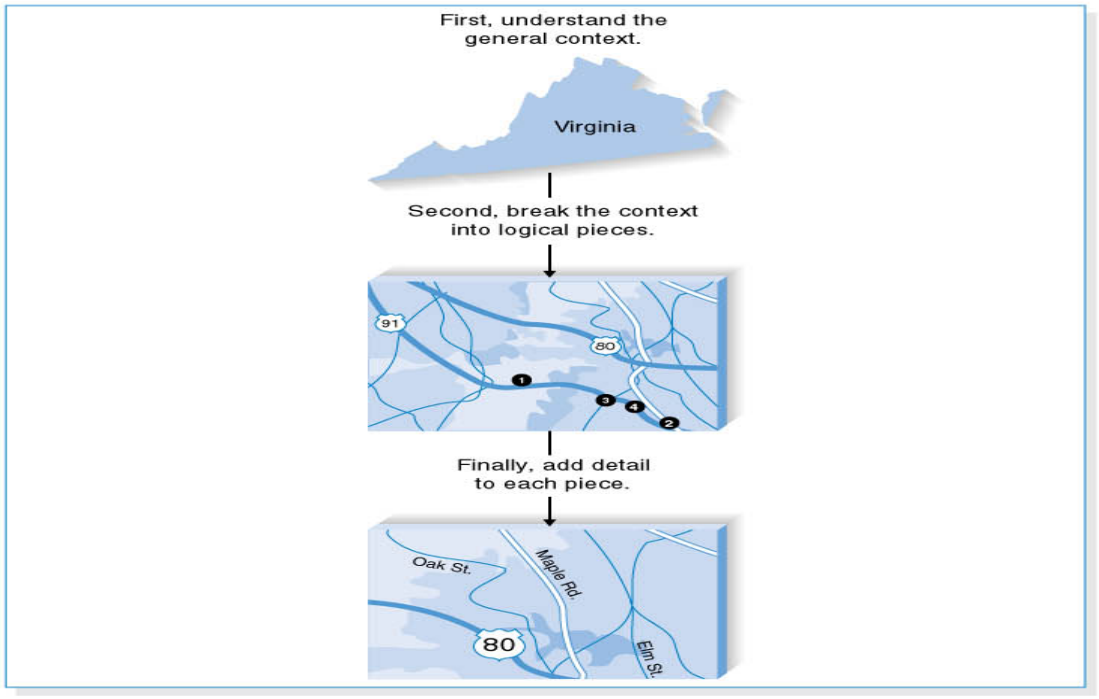
The Physical Data Flow Diagram (The How)

مخطط تدفق البيانات المادية (كيف)



Using a Top-Down Modular Approach

استخدام نهج وحدات من أعلى إلى أسفل



The Structure Chart

Important program design technique

Shows all components of code in a hierarchical format

Sequence (in what order components are invoked)

Selection (under what condition a module is invoked)

Iteration (how often a component is repeated)

مخطط الهيكل

تقنية تصميم البرنامج الهامة

لعرض كافة مكونات التعليمات البرمجية بتنسيق هرمي

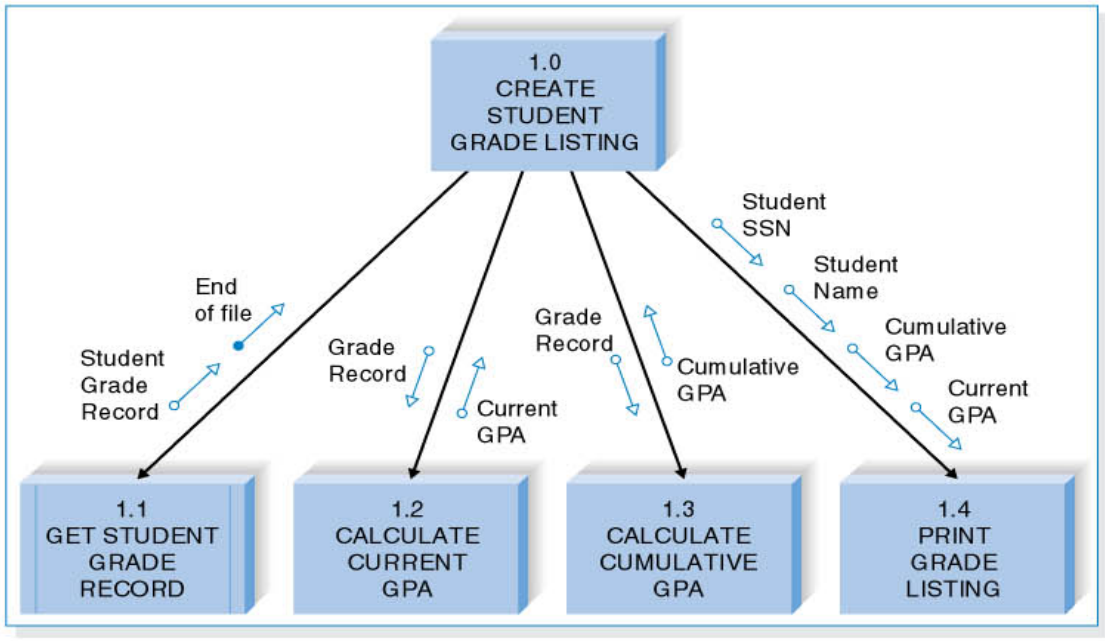
التسلسل (في ما يتم استدعاء مكونات النظام)

التحديد (تحت أي شرط يتم استدعاء وحدة نمطية)

التكرار (عدد المرات التي يتكرر فيها المكون)

Structure Chart Example

مثال على مخطط الهيكل



Structure Chart Elements

عناصر المخطط الهيكلي

Structure Chart Element	Purpose	Symbol
Every *module*: • Has a number • Has a name • Is a control module if it calls other modules below it • Is a subordinate module if it is controlled by a module at a higher level	Denotes a logical piece of the program	1.2 CALCULATE CURRENT GPA
Every *library module* has: • A number • A name • Multiple instances within a diagram	Denotes a logical piece of the program that is repeated within the structure chart	1.1 GET STUDENT GRADE RECORD
A *loop*: • Is drawn using a curved arrow • Is placed around lines of one or more modules that are repeated	Communicates that a module(s) is repeated	
A *conditional line*: • Is drawn using a diamond • Includes modules that are invoked based on some condition	Communicates that subordinate modules are invoked by the control module based on some condition	
A *data couple*: • Contains an arrow • Contains an empty circle • Names the type of data that is being passed • Can be passed up or down • Has a direction that is denoted by the arrow	Communicates that data is being passed from one module to another	grade record
A *control couple*: • Contains an arrow • Contains a filled-in circle • Names the message or flag that is being passed • Should be passed up, not down • Has a direction that is denoted by the arrow	Communicates that a message or a system flag is being passed from one module to another	end of file
An *off-page connector*: • Is denoted by the hexagon • Has a title • Is used when the diagram is too large to fit everything on the same page	Identifies when parts of the diagram are continued on another page of the structure chart	PRINT GRADE LISTING
An *on-page connector*: • Is denoted by the circle • Has a title • Is used when the diagram is too large to fit everything in the same spot on a page	Identifies when parts of the diagram are continued somewhere else on the same page of the structure chart	PRINT GRADE LISTING

Building the Structure Chart

Processes in the DFD tend to represent one module on the structure chart

Afferent processes – provide inputs to system

Central processes – perform critical system operations

Efferent processes – handle system outputs

The DFD leveling can correspond to the structure chart hierarchy (e.g., the process on the context-level DFD would correspond to the top module on the structure chart)

بناء المخطط الهيكلي

تميل العمليات في DFD لتمثيل وحدة نمطية واحدة على المخطط الهيكلي
عمليات متأصلة - تقديم مدخلات إلى النظام
العمليات المركزية - تنفيذ عمليات النظام الحرجة
عمليات فعالة - التعامل مع مخرجات النظام
يمكن أن تتطابق تسوية DFD مع التسلسل الهرمي للهيكل الهيكلي (على سبيل المثال، العملية على مستوى DFDعلى مستوى السياق ستتوافق مع الوحدة النمطية الأعلى على المخطط الهيكلي)

Types of Structure Charts

Transaction structure – control module calls subordinate modules, each of which handles a particular transaction

Few afferent processes

Many efferent processes

Higher up levels of structure chart

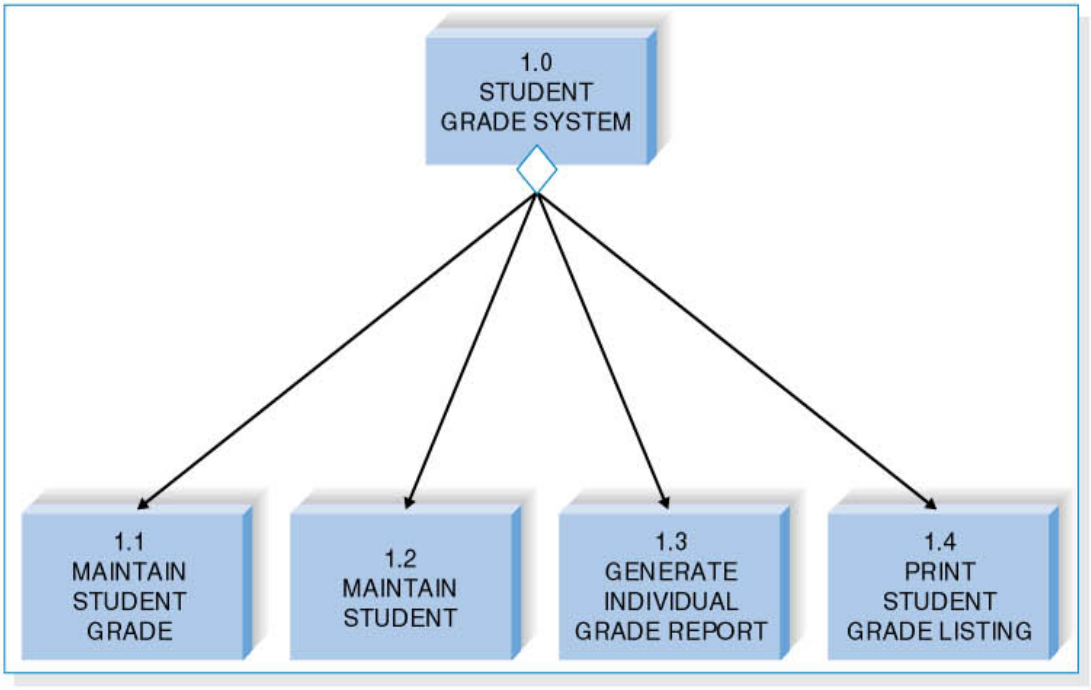
Using inputs to create a new output

أنواع المخططات الهيكلية

هيكل المعاملات - وحدة التحكم يدعو الوحدات الفرعية، كل منها يعالج معاملة معينة
عدد قليل من العمليات
العديد من العمليات إيفرنت
ارتفاع مستويات أعلى من هيكل الرسم البياني
استخدام المدخلات لإنشاء مخرجات جديدة

Transaction Structure

هيكل المعاملة



Transform Structure

This structure has a control module that calls several subordinate modules in sequence after which something "happens."

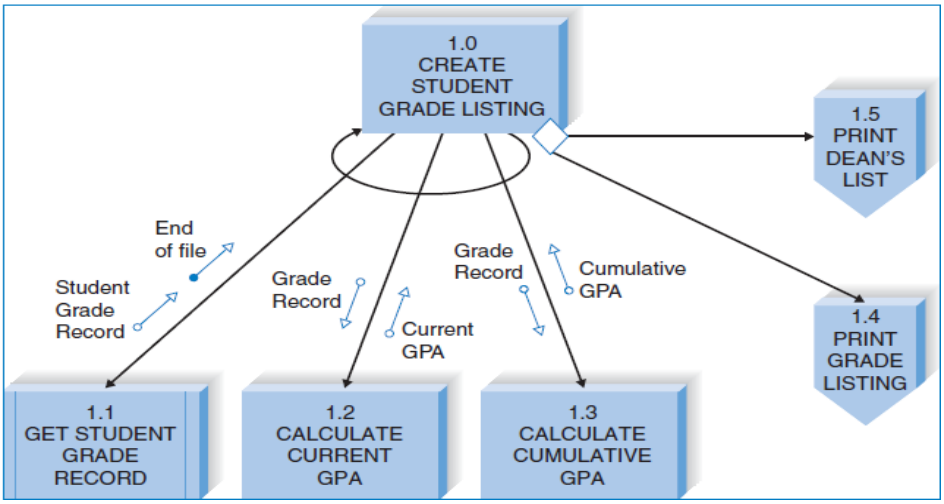
These modules are related because together they form a process that transforms some input into an output

هيكل التحويل

هذا الهيكل لديه وحدة التحكم التي تدعو العديد من الوحدات الثانوية في تسلسل بعد ذلك شيء "يحدث".
ترتبط هذه الوحدات لأنها معا تشكل عملية تحول بعض المدخلات إلى مخرجات

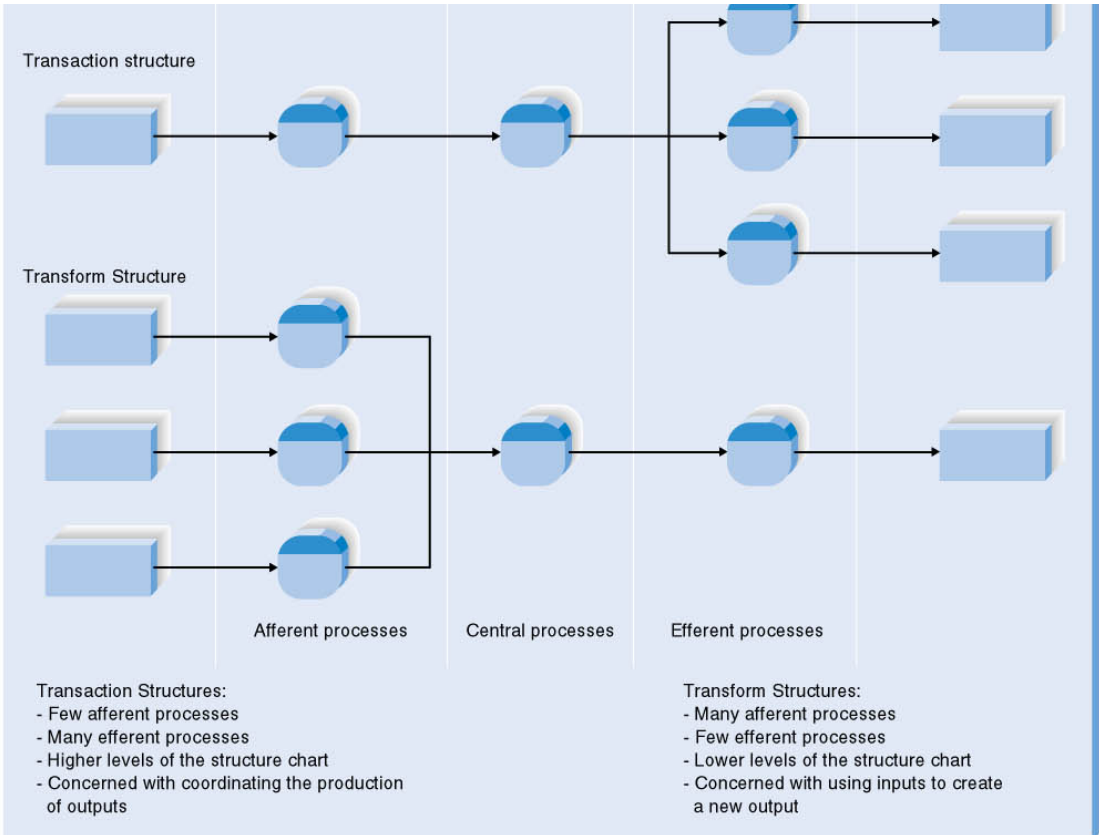
Transform Structure

هيكل التحويل



Transform versus Transaction Structures

التحويل مقابل هياكل المعاملات



Steps in Building the Structure Chart

خطوات بناء الهيكل خريطة

1-Identify top level modules and decompose them into lower levels

2-Add control connections

3-Add couples

4-Review and revise again and again until complete

1-تحديد وحدات المستوى الأعلى وتحللها إلى مستويات أقل
2-إضافة اتصالات التحكم
3-إضافة الأزواج
4-مراجعة وتنقيح مرارا وتكرارا حتى كاملة

Design Guidelines

High quality structure charts result in programs that are modular, reusable and easy to implement.

Measures include:

Cohesion

Coupling

Appropriate levels of fan-in and fan-out

موجهات التصميم
عالية الجودة هيكل المخططات يؤدي إلى البرامج التي هي وحدات، يمكن إعادة استخدامها وسهلة التنفيذ.
وتشمل التدابير:
تماسك
اقتران
مستويات مناسبة من مروحة في والمروحة التدريجي

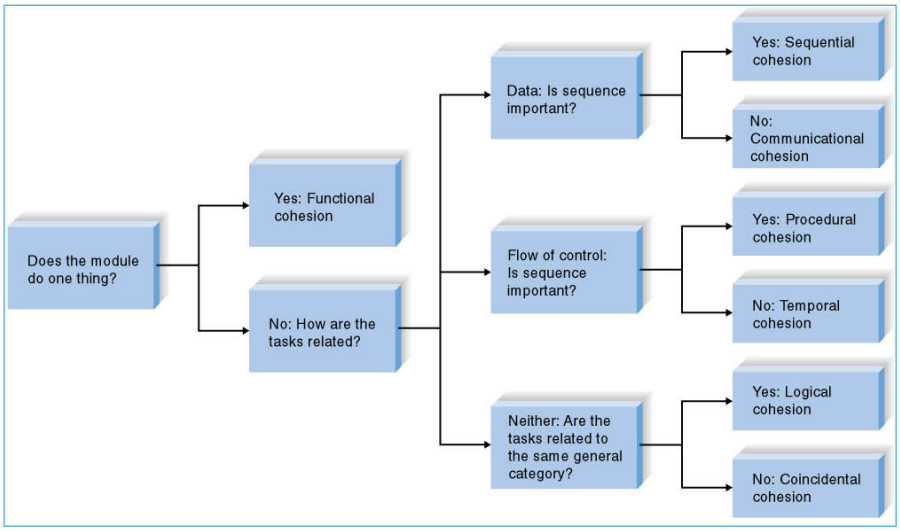
Types of Cohesion (GPA = grade point average)

أنواع التماسك (GBA = درجة الصف معدل)

	Type	Definition	Example
Good	Functional	Module performs one problem-related task.	Calculate Current GPA The module calculates current GPA only.
	Sequential	Output from one task is used by the next.	Format and Validate Current GPA Two tasks are performed, and the formatted GPA from the first task is the input for the second task.
	Communicational	Elements contribute to activities that use the same inputs or outputs.	Calculate Current and Cumulative GPA Two tasks are performed because they both use the student grade record as input.
	Procedural	Elements are performed in sequence but do not share data.	Print Grade Listing The module includes the following: housekeeping, produce report.
	Temporal	Activities are related in time.	Initialize Program Variables Although the tasks occur at the same time, each task is unrelated.

Cohesion Decision Tree (Adopted From Page-Jones, 1980)

تماسك شجرة القرار (اعتمد من صفحة-جونز، 1980)



Factoring

-Process of dealing with “low" cohesion

-Separates tasks into different modules

-Reduces use of control flags

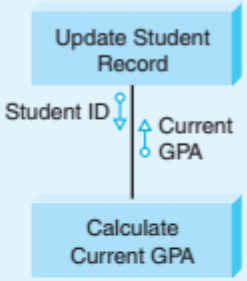
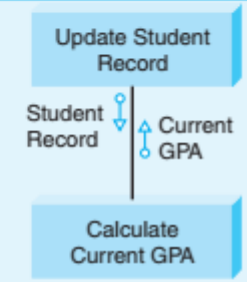
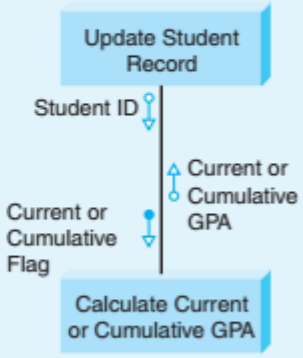
العوملة

- عملية التعامل مع التماسك "المنخفض"

-يفصل المهام إلى وحدات مختلفة

- يقلل من استخدام أعلام التحكم

Types of Coupling(أنواع من اقتران)

Type		Definition	Example
Good	Data	Modules pass fields of data or messages.	 All couples that are passed are used by the receiving module.
	Stamp	Modules pass record structures.	 Not all of the student record is used by the receiving module; only the *student ID* field is.
	Control	Module passes a piece of information that intends to control logic.	 The receiving module has to determine which GPA to calculate.
	Common	Modules refer to the same global data area.	Typically, common coupling cannot be shown on the structure chart; it occurs when modules access the same data areas, and errors made in those areas can ripple through all the modules that use the data.
Bad	Content	Module refers to the inside of another module.	Module A: Update Student If student = new Then go to Module B Module B: Create Student At all costs, avoid modules referring to each other in this way.

Your Turn

What, if anything, happens to coupling when you create modules that are more cohesive?

What, if anything happens to the cohesiveness of modules when you lower the coupling among them?

دورك

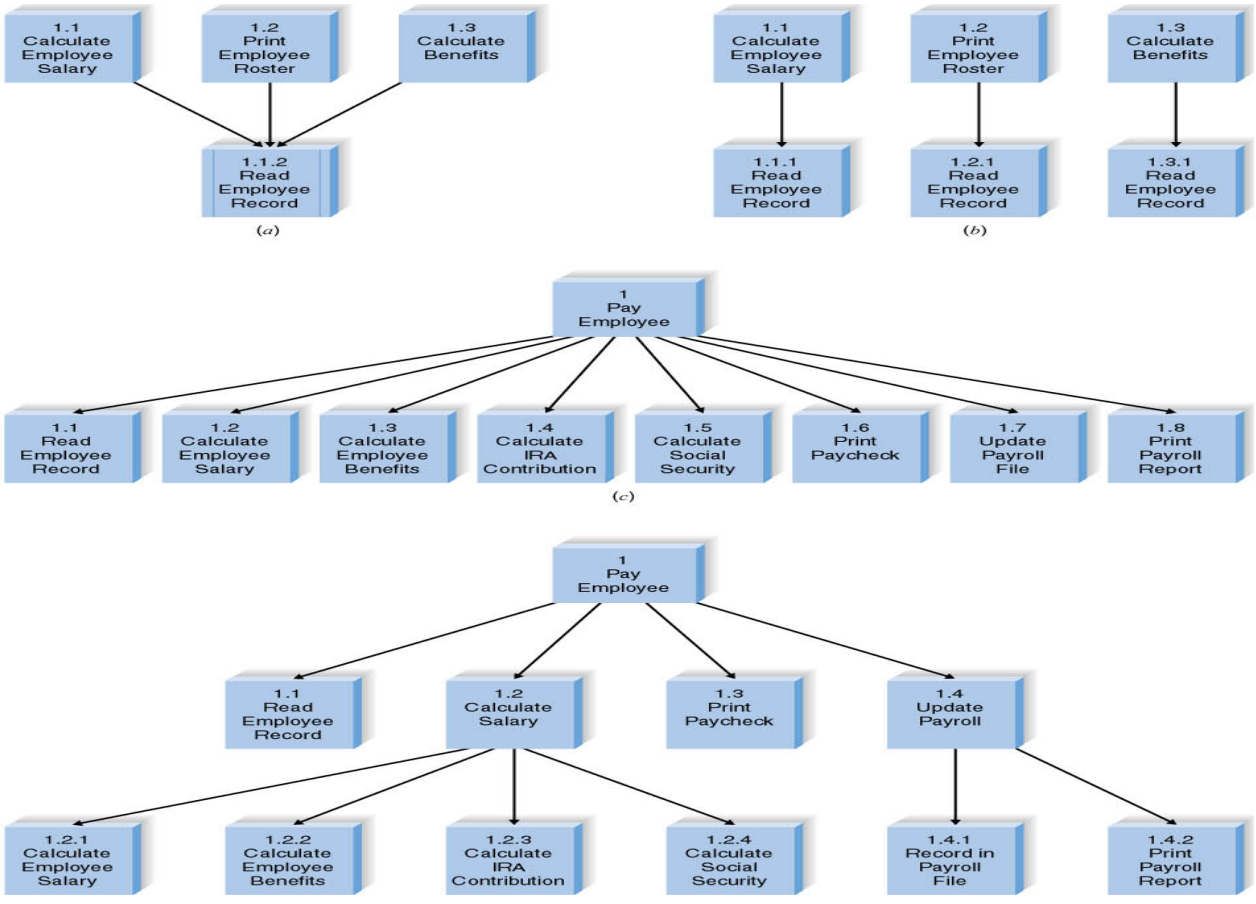
ماذا يحدث، إذا كان هناك أي شيء، للاقتران عند إنشاء وحدات أكثر تماسكا؟

ماذا، إذا حدث أي شيء لتماسك وحدات عند خفض اقتران فيما بينها؟

Examples of Fan-in and Fan-out

توجيه اشارات لمستقبل واحد = Fan-in

توجيه اشاره الى عدة مستقبلين = fan-out



Quality Checklist

1-Library modules have been created where ever possible

2-The diagram has a high fan-in structure

3-Control modules have no more than 7 subordinates

4-Each module performs only one function (high cohesion)

5-Modules sparingly share information (loose coupling)

6-Data couples that are passed are actually used by the accepting module

7-Control couples are passed from “low to high"

8-Each module has a reasonable amount of code associated with it

قائمة مرجعية للجودة

1-تم إنشاء وحدات المكتبة حيثما كان ذلك ممكنا
2-الرسم البياني لديه مروحة عالية في هيكل
3-وحدات التحكم لديها ما لا يزيد عن 7 المرؤوسين
(4-كل وحدة أداء وظيفة واحدة فقط (التماسك عالية
5-وحدات تتشارك بشكل متقاسم المعلومات (اقتران فضفاض)
6-الأزواج البيانات التي يتم تمريرها تستخدم فعلا من قبل وحدة قبول
7-تحكم الأزواج يتم تمريرها من "منخفضة إلى عالية"
8-كل وحدة لديها كمية معقولة من التعليمات البرمجية المرتبطة به

Program Specifications Content

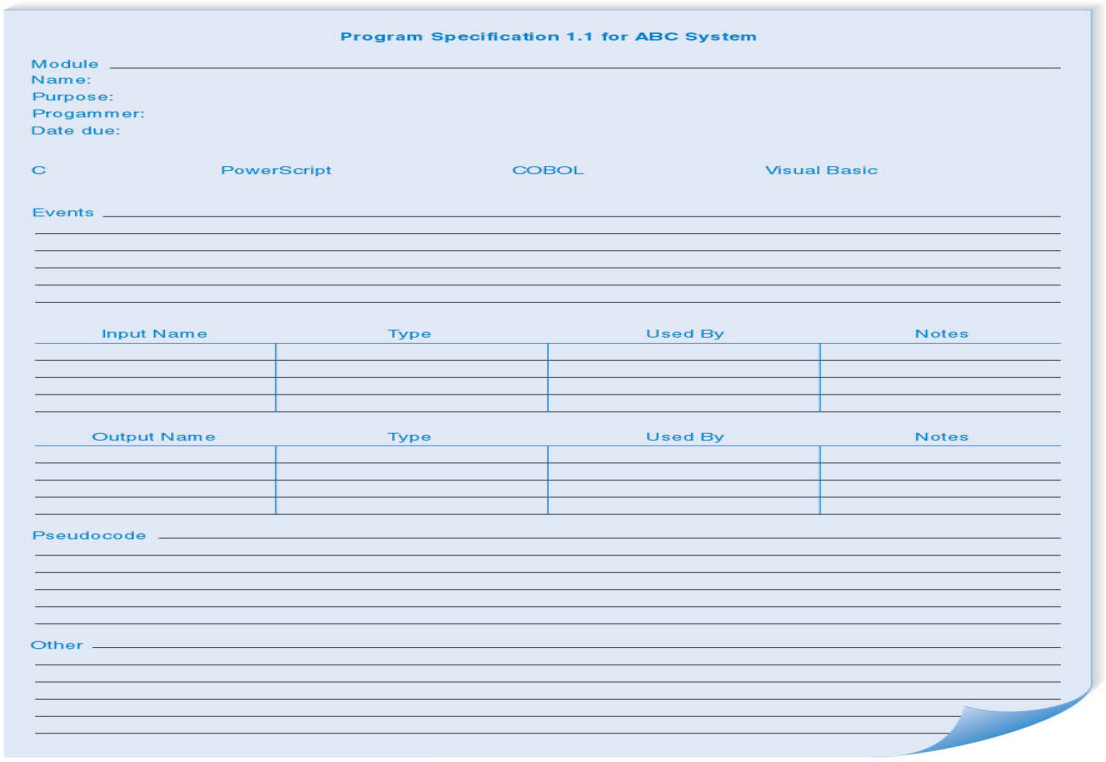
No standard approach
Include program information
Note events that trigger actions
List inputs and outputs
Include pseudocode
Present additional notes and comments

مواصفات البرنامج المحتوى

لا يوجد نهج قياسي
تضمين معلومات البرنامج
لاحظ الأحداث التي تؤدي الإجراءات
قائمة المدخلات والمخرجات
تضمين بسيودوكود
تقديم ملاحظات وتعليقات إضافية

Program Specification Form

نموذج مواصفات البرنامج



Program Specification 1.1 for ABC System

Module Name:
Purpose:
Progammer:
Date due:

C PowerScript COBOL Visual Basic

Events

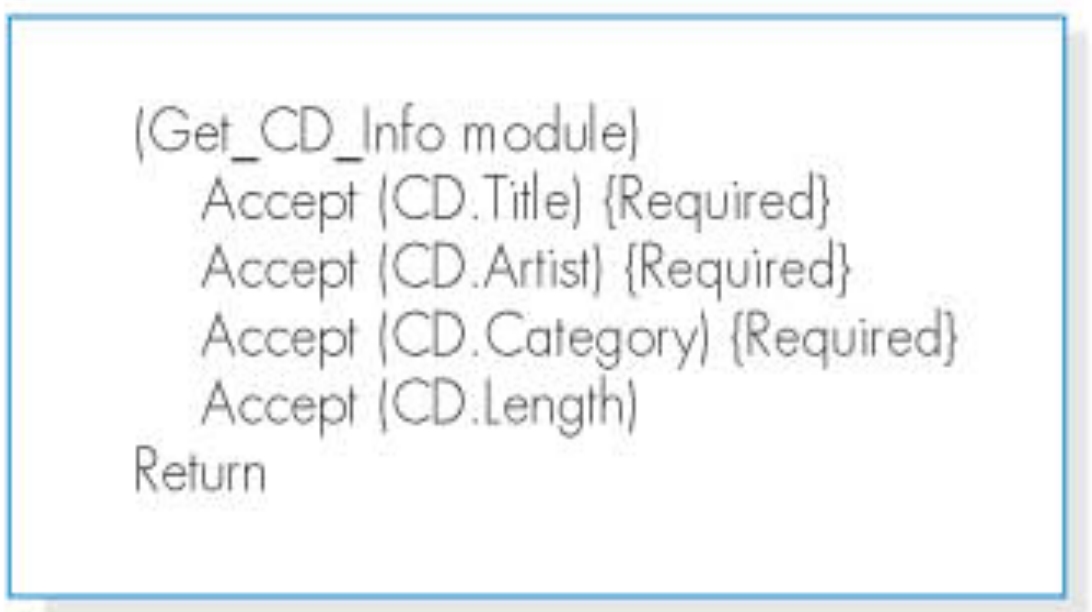
Input Name	Type	Used By	Notes

Output Name	Type	Used By	Notes

Pseudocode

Other

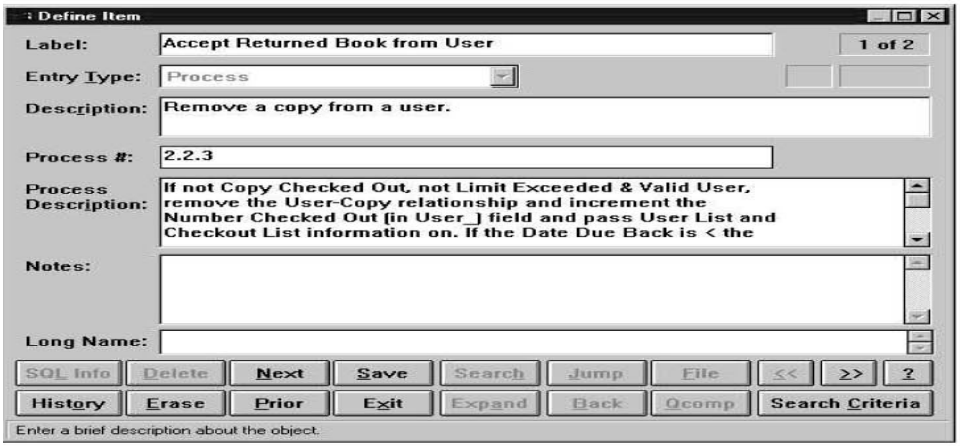
Pseudocode Example



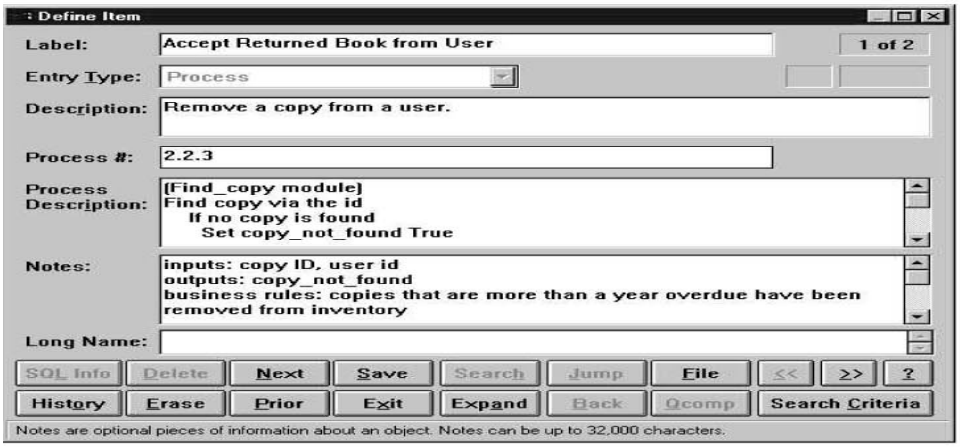
```
(Get_CD_Info module)
    Accept (CD.Title) {Required}
    Accept (CD.Artist) {Required}
    Accept (CD.Category) {Required}
    Accept (CD.Length)
Return
```

Process Description – Analysis and Design

وصف العملية - التحليل والتصميم



Analysis Phase Process Specification with structured English process description



Design Phase Process Specification:

- structured English has been changed to pseudocode
- relevant specification information like inputs, outputs, and business rules have been added to the Notes section

By : Shadia Alsolami & Nouf Binoryan

good luck for all