# Software Engineering

## U-02 系統工程

Neter Chao Ph.D.

# Agenda

- 系統工程四維構面(層級、耦合、需求與時間)
- 瞭解系統中的軟體為何會受到系統工程問題的影響
- 瞭解突顯的系統特性之概念，例如可靠性、執行效能、安全與保護等
- 瞭解系統設計過程中為何需要考慮系統的環境

# 系統工程

# 何謂「系統」？

- 系統是由一群相關的組成元件集合而成，為了達成某個目的而共同運作
- 系統可以包含軟體以及可讓人員操作的機械、電機與電子式的硬體
- 系統的組成元件必須依賴其他系統組成元件
- 系統各組成元件的性質與行為不可避免的必須相互混合

# 系統工程的問題

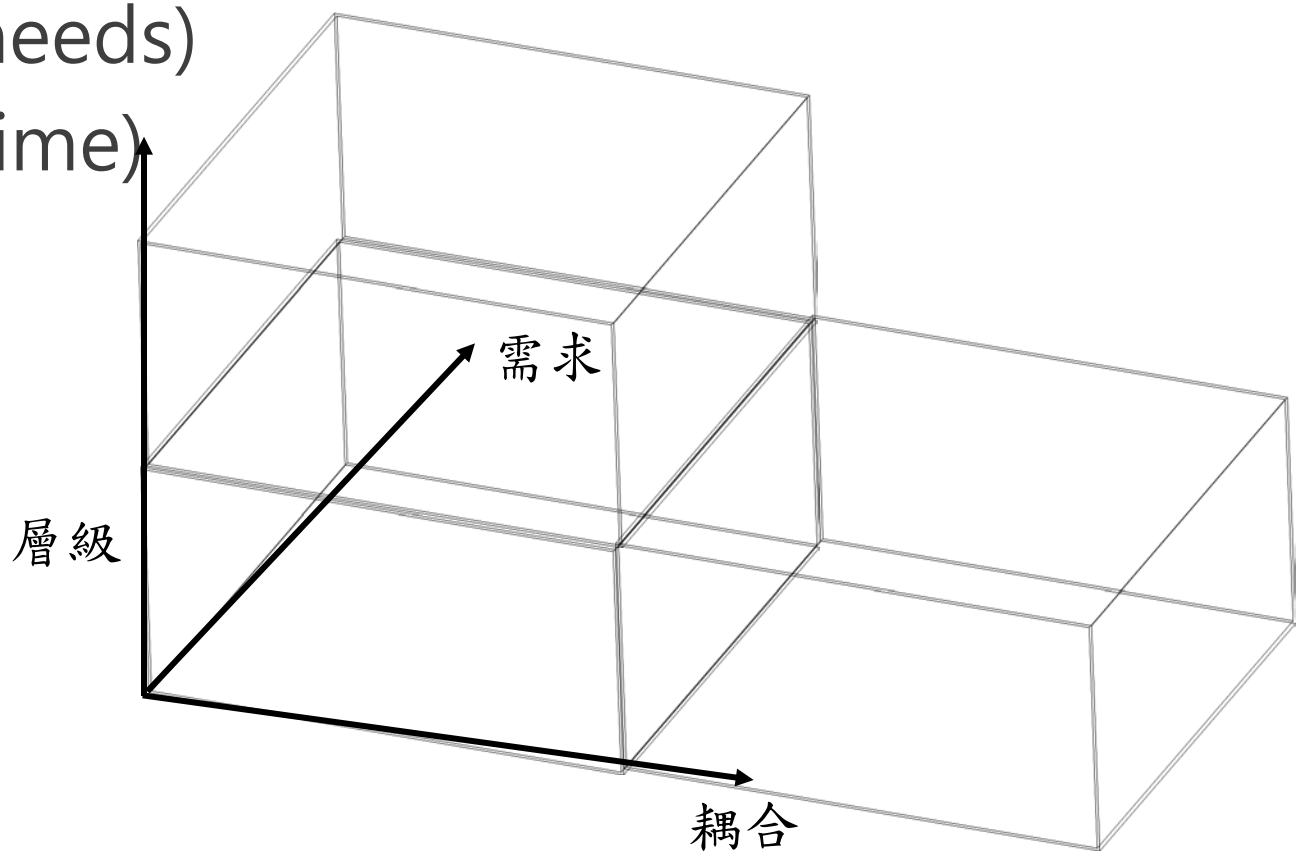
- 大型系統通常設計用來解決「複雜又難以處理的問題」
- 系統工程需要許多跨不同學科的協調
  - 幾乎不可避免的必須在不同的組成元件之間做取捨
  - 不同工程領域之間通常缺乏互信與瞭解
- 系統必須設計成可以在變更的環境中持續運作許多年

# 軟體與系統工程

- 系統中軟體部分的份量愈來愈重。以軟體驅動的通用電子事產品已漸漸取代特殊用途的系統
- 系統工程的問題與軟體工程的問題類似
- 不幸的是軟體在系統工程中被視為是問題。許多大型系統專案都是由於軟體的問題而延遲

# 系統工程四維構面

- 層級(hierarchical )
- 耦合(coupling )
- 需求(needs)
- 時間(time)



# 突顯性質(Emergent Properties)

- 突顯性質不是系統中某個組成元件的特性，而是當系統以整體來考量時所出現的性質
- 突顯性質是系統元件之間的關係所形成的結果
- 因此，這些性質只有在各元件整合成一個系統時才可以進行評估與度量

# 突顯性質的範例

- ***系統的整體價值***
  - 這個性質可以從個別組成元件的性質計算而來 。
- ***系統的可靠性***
  - 這個性質必須根據系統組成元件的可靠性以及各元件之間的關係而定 。
- ***系統的可使用性***
  - 這是一個非常複雜的性質，它不是直接從系統的軟硬體而來，而是根據系統的操作人員和使用環境而定。

# 突顯性質的類型

- **功能性的性質**
  - 當系統的所有組成部分一起運作而達成某個目標時所出現的性質。例如，以各種零件組合而成的自行車，在組合完成之後就具有一項能夠當成運輸工具的功能性質。
- **非功能性的性質**
  - 例如可靠性、執行效能、安全性和保全性。這些性質都跟操作環境中的系統行為有關。而且對電腦化的系統而言這些是非常重要的性質，因為只要系統無法達成定義的最小等級性質，系統就會被視為無法使用。

# 系統可靠度工程

- 由於元件相互之間的相依性，使得錯誤會在系統中擴散開來
- 系統故障通常是由於沒有預見到元件之間的相互關係所產生的
- 各元件的所有可能關係不太可能都預期得到
- 軟體可靠度的度量可能會誤導系統的可靠度

# 可靠度的影響

- ***硬體可靠度***
  - 指硬體元件發生故障的可能機率，以及修護該元件所需的時間。
- ***軟體可靠度***
  - 指軟體元件產生錯誤結果的可能性。軟體故障通常不同於硬體的故障，因為軟體不會被用壞。
- ***作業員可靠度***
  - 指系統操作人員造成失誤的可能性。

# 可靠度的關係

- 硬體故障可能會引發假訊號的產生，這些訊號則非軟體預期的輸入資料範圍
- 軟體錯誤可能會觸動警報，因而造成作業人員的壓力，導致作業人員出錯
- 系統安裝的所在環境也會影響到它的可靠度

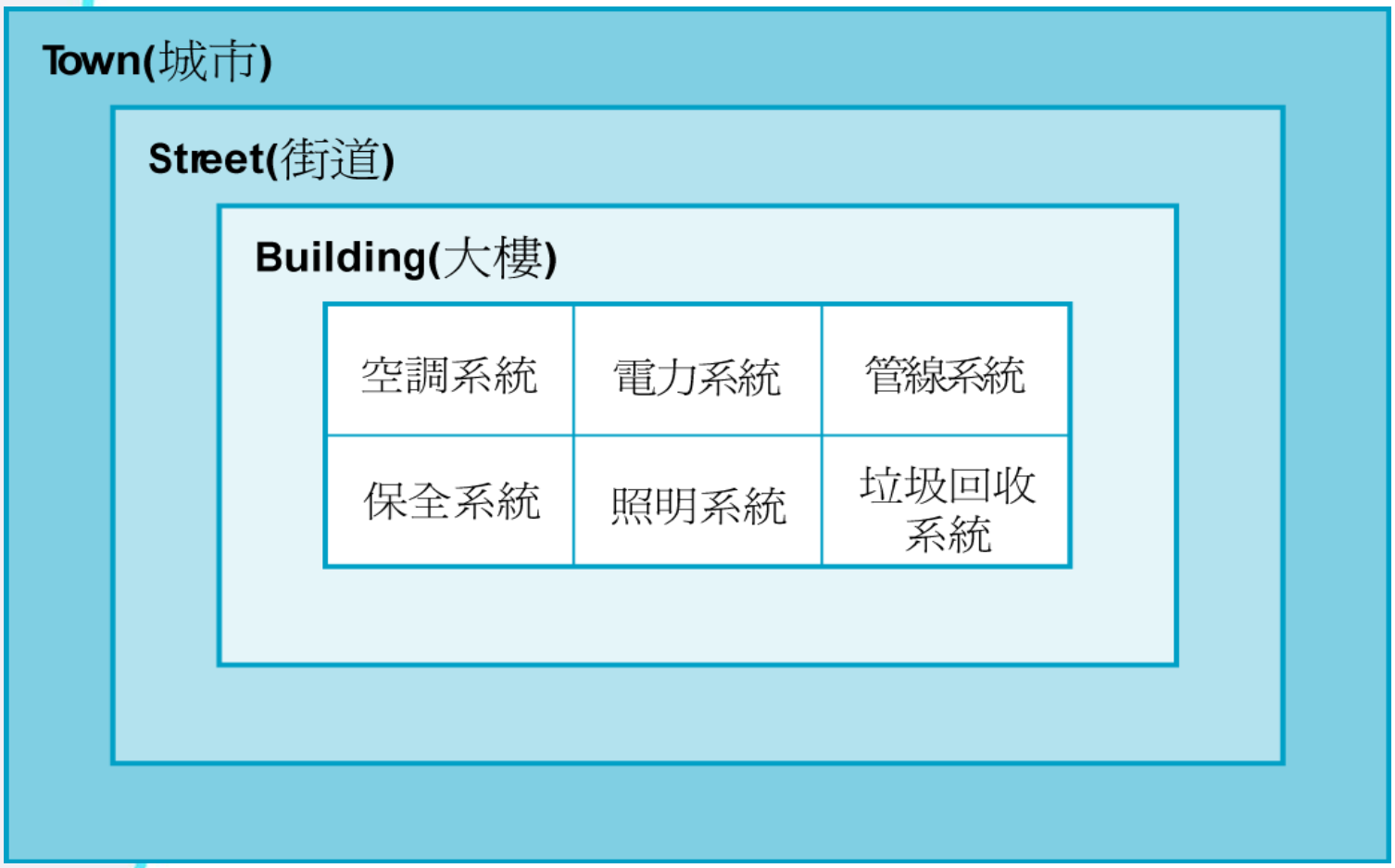
# 「不應該」的性質

- 執行效能和可靠度之類的性質是可以被度量的
- 然而，有些性質是以系統不應該出現的性質來表示，例如
  - 安全性(Safety) – 系統不應該出現不安全的行為
  - 保全性(Security) – 系統不應該允許未經授權的使用者使用
- 度量或評估這些性質非常困難

# 系統與它們的環境

- 系統並不是獨立的實體，而是存在某一個環境中
- 系統的功能可能會改變它的環境
- 環境也會影響到系統的功能，例如：系統可能需要從它的環境取得電力的支援
- 組織與實體的環境也是很重要的

# 系統階層架構圖


Town(城市)
Street(街道)
Building(大樓)
空調系統
電力系統
管線系統
保全系統
照明系統
垃圾回收系統

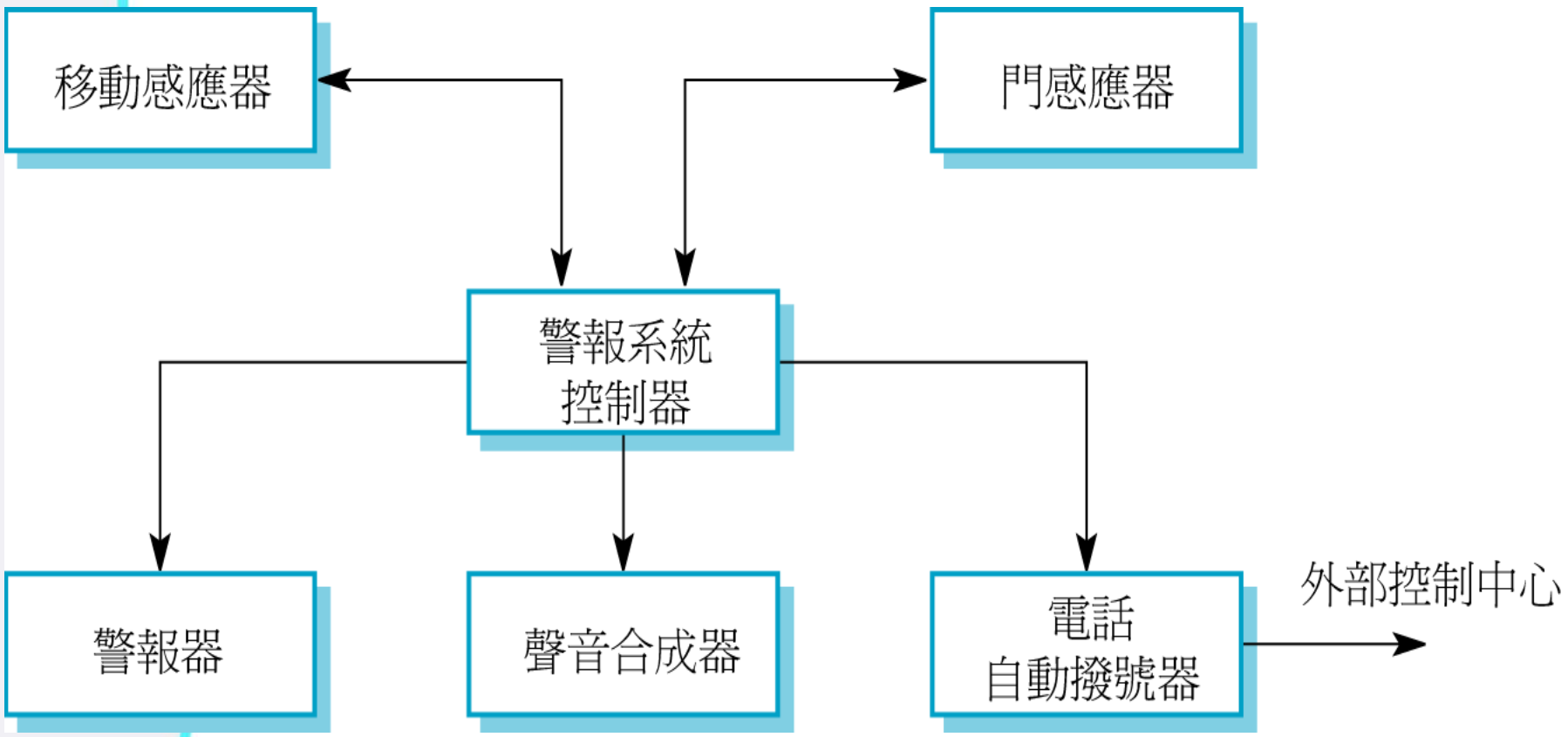
# 人與組織的因素

- *程序的改變*
  - 系統是否需要改變，以符合環境中的工作程序？
- *工作的改變*
  - 系統是否會造成環境中使用者技能的降低，或是造成工作方式的改變？
- *組織的改變*
  - 系統是否會改變組織中政治的權力結構？

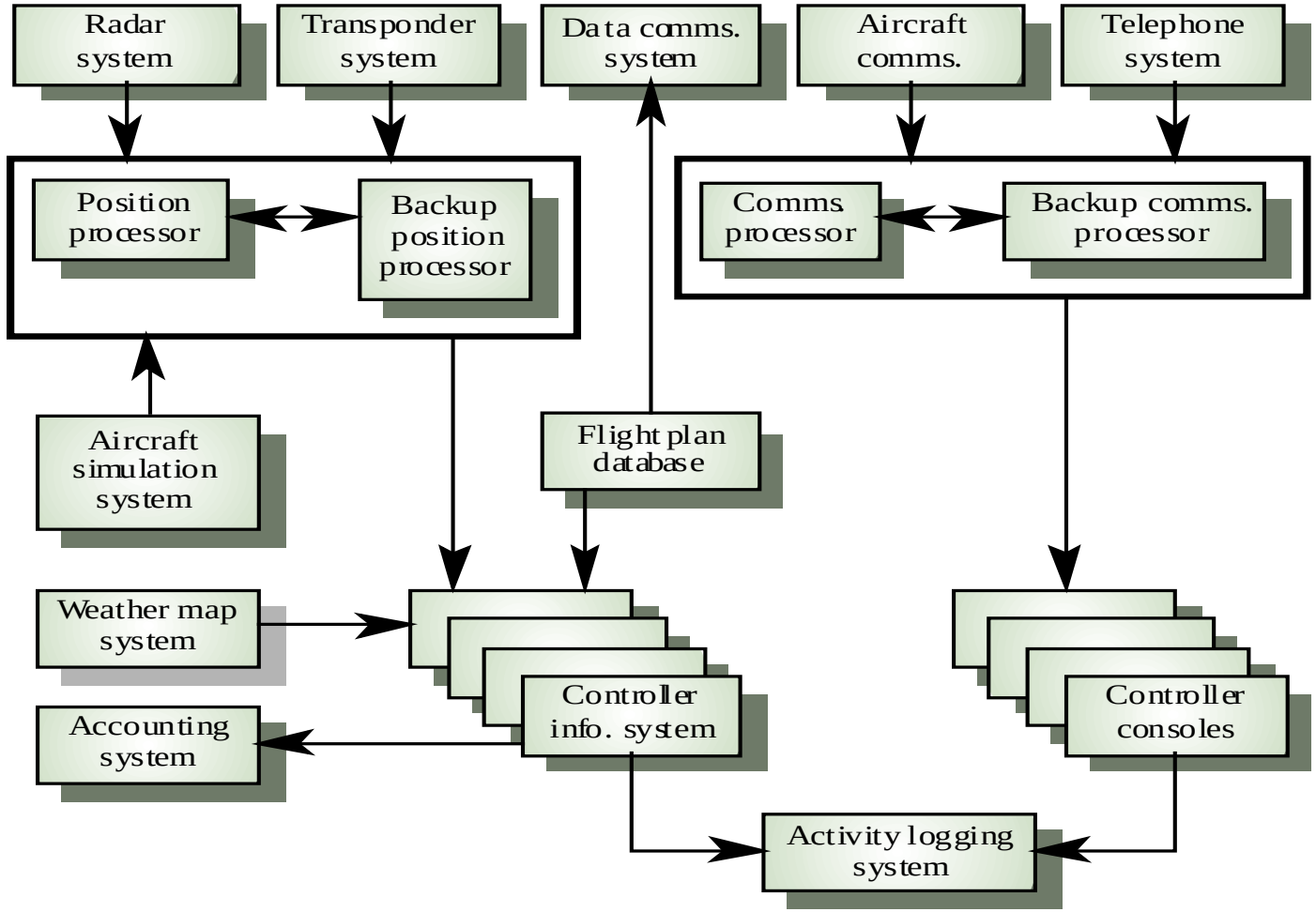
# 建立系統架構模型

- 架構模型可以表示組成系統的各個子系統的抽象檢視
- 模型中可以包含子系統之間的主要資訊流
- 模型通常是以方塊圖來表示
- 可以分辨出模型中不同類型的功能元件

# 侵入者警報系統



# ATC 系統架構


Radar system
Transponder system
Data comms. system
Aircraft comms.
Telephone system
Position processor
Backup position processor
Comms. processor
Backup comms. processor
Aircraft simulation system
Flight plan database
Weather map system
Accounting system
Controller info. system
Controller consoles
Activity logging system

# 功能性的系統元件

- 感應元件
- 觸動元件
- 運算元件
- 通訊元件
- 協調元件
- 介面元件

# 系統元件

- 感應元件
  - 用來收集系統的環境資訊，例如航管系統中的雷達
- 觸動元件
  - 能夠造成系統環境某些改變的元件，例如開啟和關閉或開大或減少管線中水流量的活門
- 運算元件
  - 能夠接受輸入，再根據輸入進行某些運算，進而產生某些輸出的元件，例如可執行實數運算的浮點數處理器

# 系統元件

- 通訊元件
  - 可讓系統中其他元件相互溝通的元件，例如可連結大樓中各個電腦的 Ethernet 連線
- 協調元件
  - 用來協調其他元件運作情形的系統元件，例如即時系統中的排程程式
- 介面元件
  - 提供與其他系統組成元件互動的功能，例如作業員的介面
- 所有組成元件現在通常是由軟體來控制

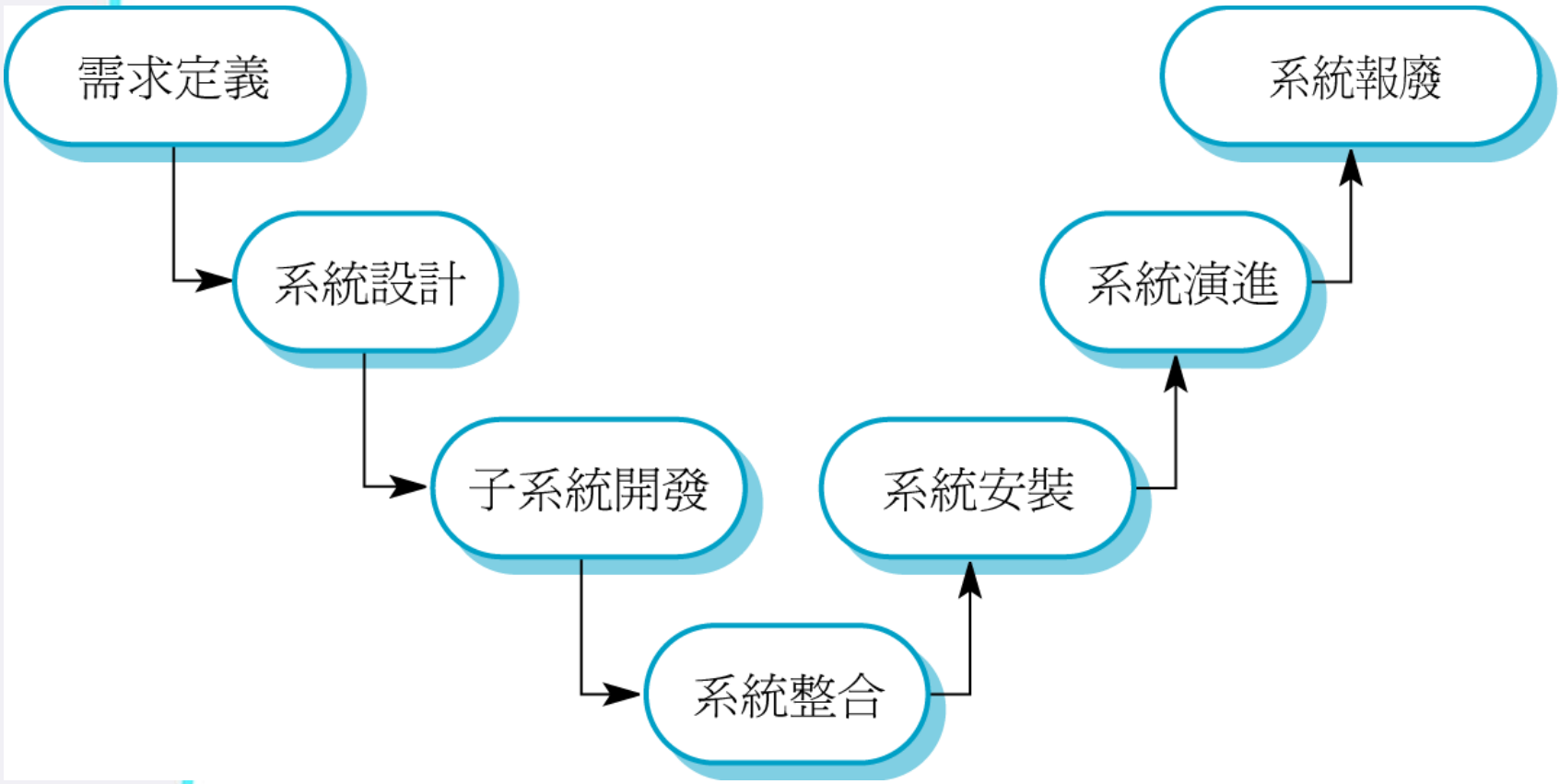
# 警報系統中的元件類型

- 感應器
  - 移動感應器、房門感應器
- 觸動器
  - 警報器
- 通訊
  - 電話自動撥號器
- 協調
  - 警告控制器
- 介面
  - 聲音合成器

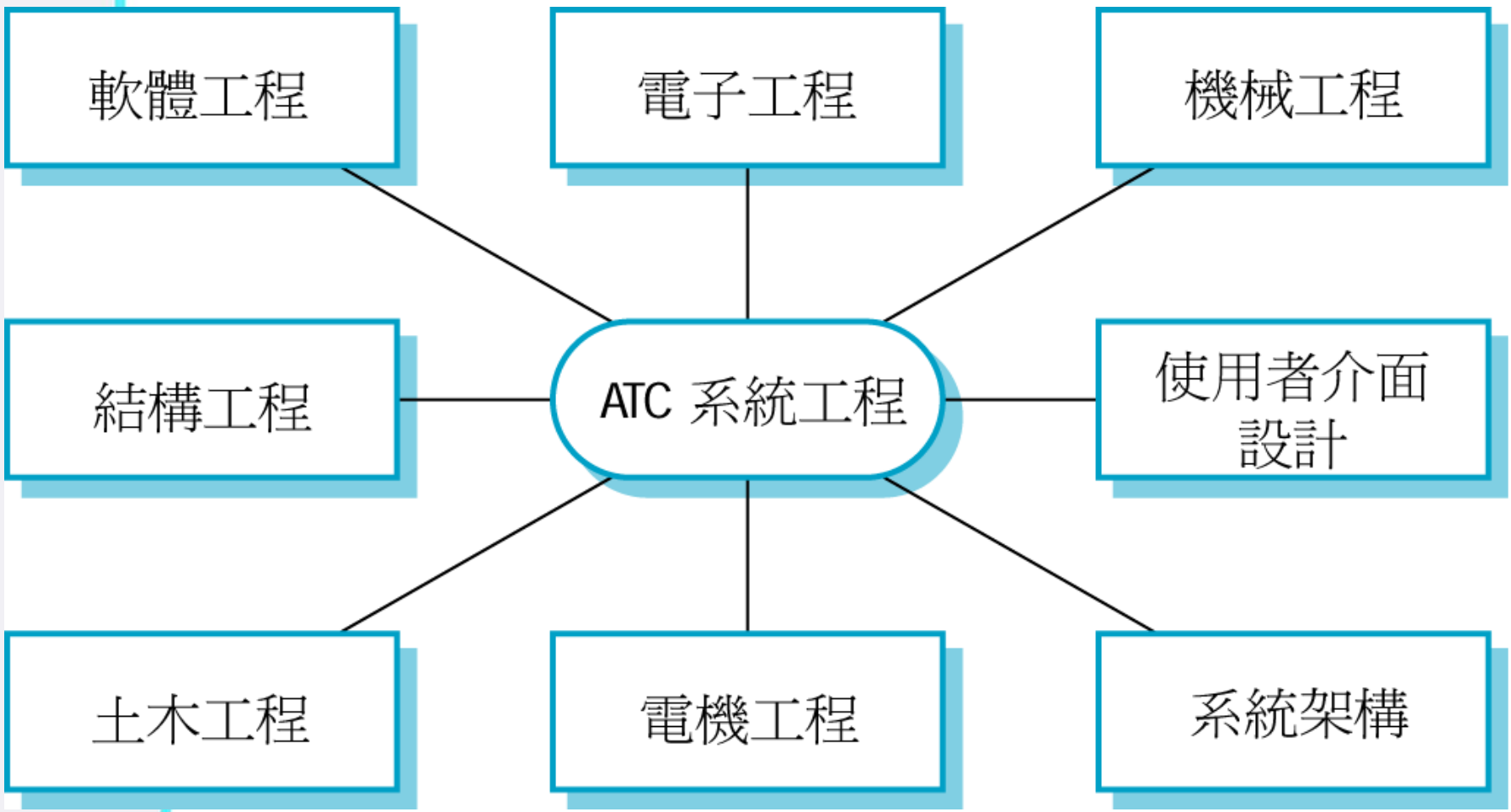
# 系統工程的程序

- **因為系統不同部分需要並行的開發，所以通常是以「瀑布式」模型為主**
  - 因為硬體的變更非常昂貴，所以不同階段之間的重複範圍很小。軟體可以必須補償硬體發生的問題
- **必須有不同領域的工程師參與一起合作**
  - 產生的誤解可能會變大，不同領域通常會使用不同的辭彙，因此需要大量的溝通與協調。工程師們可能也會有各自的工作需要完成

# 系統工程程序



# 不同學科領域的參與



# 系統需求定義

- 這個階段需要定義下列三種需求：
  - 抽象的功能需求 。以抽象方式定義系統功能
  - 系統的性質 。定義通用的系統非功能性需求
  - 系統不應該有的特性。指定不可接受的系統行為
- 定義系統的整體組織目標

# 系統目標

- **功能性目標**
  - 為辦公大樓提供一個防火和入侵者警報系統，以便提供內部和外部的火警警告或未經授權的侵入警告。
- **組織性目標**
  - 確保大樓內各項工作的正常運作，不受意外事件的侵擾，例如火警和未經授權的侵入行為。

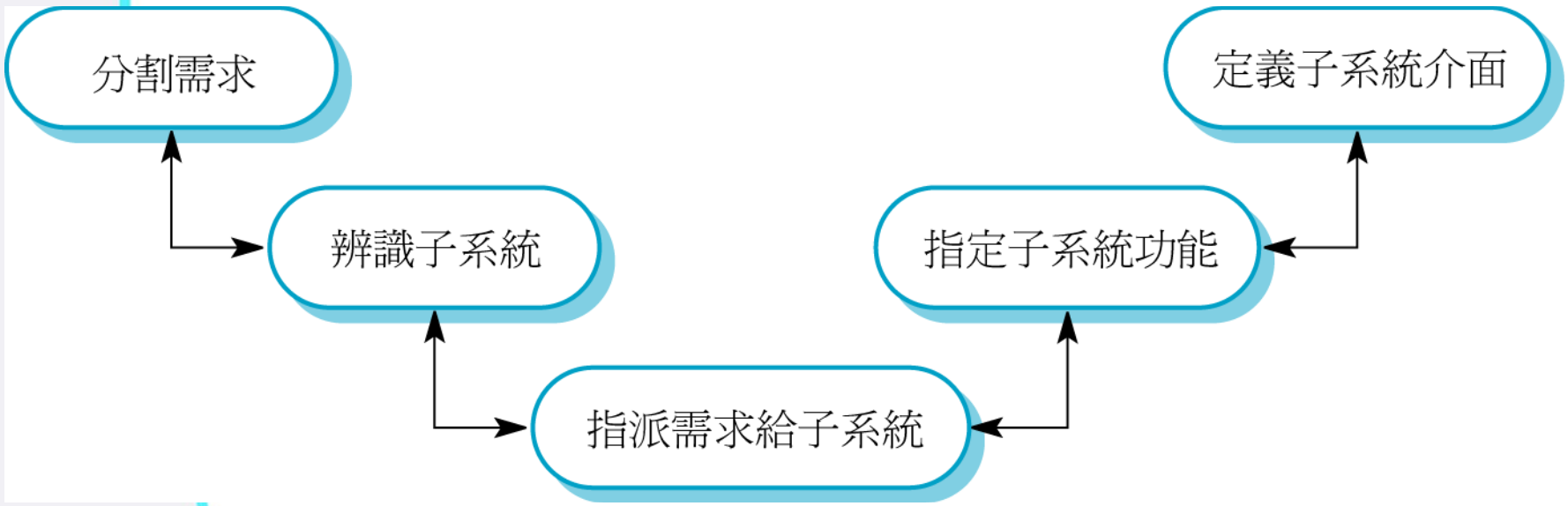
# 系統需求問題

- 系統在指定期間又有變更
- 必須預期系統使用壽命期間的硬體與通訊的開發
- 若沒有系統元件結構的印象，難以定義它的非功能需求。

# 系統設計程序

- **分解需求**
  - 將各需求組織成相關的群組
- **辨識子系統**
  - 辨識出一組子系統，將這些子系統集合起來可以符合系統的需求
- **指派需求給子系統**
  - 當需要整合 COTS 時會造成特殊的問題
- **指定子系統功能**
- **定義子系統介面**
  - 並行子系統開發的主要活動

# 系統設計程序



# 系統設計問題

- 分解成硬體、軟體和人員組成元件的需求可能牽涉到大量的溝通與協調
- 難以設計的問題通常會假設可以很容易的用軟體來解決
- 硬體平台可能不適用於軟體的需求，所以軟體必須對此做些補償

# 子系統開發

- 典型的平行開發專案，分別進行硬體、軟體與通訊的開發
- 可能牽涉到某些現成商用系統(COTS, Commercial Off-the-Shelf)的採購
- 各個實作小組之間可能缺乏溝通
- 若系統的變更需要經過官僚機制的拖延，開發時程可能就會由於需要重新修訂而延遲

# 系統整合

- 將硬體、軟體和人員集合在一起組成一個系統的過程
- 應該以遞增的方式來處理，以便可以一次整合一個子系統
- 子系統之間的介面問題通常會在這個階段發現
- 未經協調的系統元件交付成果可能就會出現問題

# 系統安裝

- 環境的假設可能不正確
- 人員可能會抗拒新系統的引入
- 系統可能必須與其他替代系統並存一段時間
- 可能會有實體的安裝問題發生，例如佈線問題
- 必須辨識出作業員所需的訓練

# 系統運作

- 將會揭露一些未發現的需求
- 使用者可能會以系統設計者意料之外的方式來使用系統
- 可能會在與其他系統的互動中發現一些問題
  - **實體不相容的問題**
  - **資料轉換問題**
  - **因介面不一致產生的作業員錯誤率增加**

# 系統演化

- 大型系統有較長的使用壽命，為了符合需求的變更，所以必須進行演化
- 演化非常耗費成本
  - 必須從技術與商業的觀點來分析變更
  - 子系統間的互動會產生一些非預期的問題
  - 原始設計的決策通常很少有記錄
  - 系統進行變更後可能會毀損系統的結構
- 需要進行維護的現存系統有時候稱為「既有舊系統」(legacy systems)

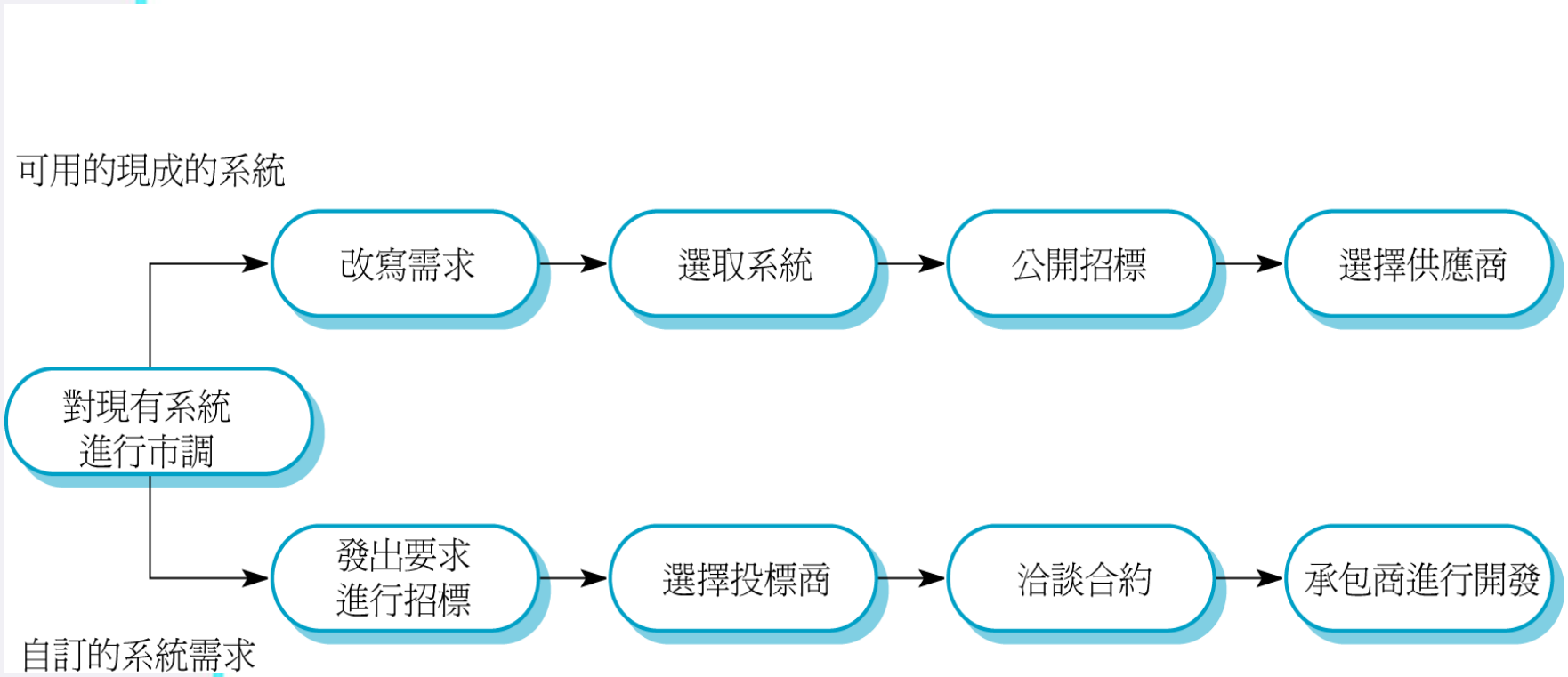
# 系統報廢

- 在系統使用壽命到期後停止系統的服務
- 可能需要一併移除污染環境的一些物質，例如危險的化學物質
  - 在系統設計時應該規劃使用封裝的方式
- 資料可能必須重新建構，或轉換成其他系統可以使用的格式

# 系統採購

- 為組織取得符合某些需求的系統
- 在進行採購之前通常必須有系統規格和架構設計
  - 必須有規格才能讓承包商進行系統的開發
  - 規格中可能允許購買現成的商用系統(COTS)，這個方式通常比從頭開發系統來得便宜

# 系統採購程序


可用的現成的系統
改寫需求
選取系統
公開招標
選擇供應商
對現有系統
進行市調
發出要求
進行招標
選擇投標商
洽談合約
承包商進行開發
自訂的系統需求

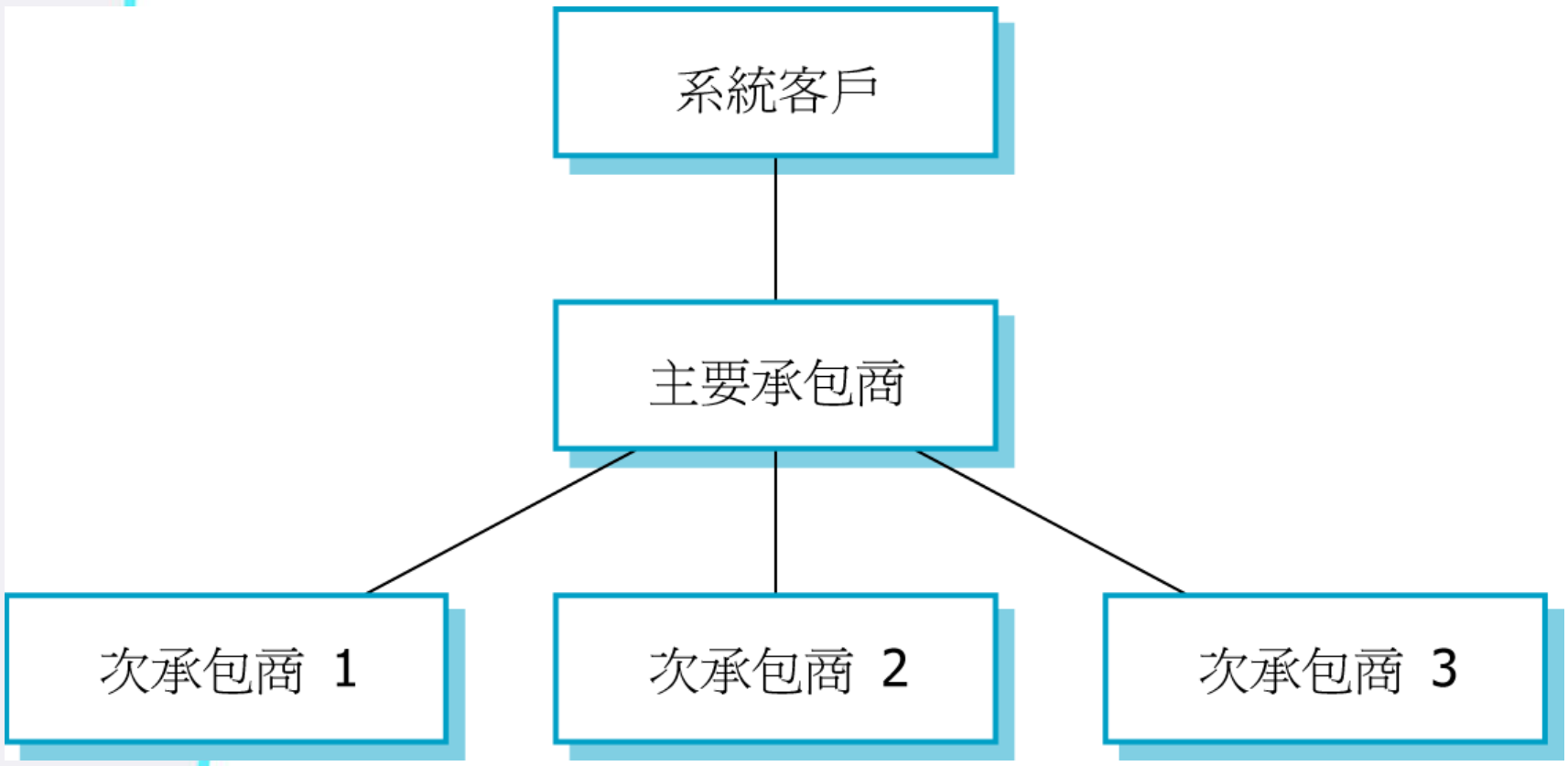
# 採購問題

- 需求可能必須修改，以符合現成軟體元件的功能
- 需求規格可能是系統開發合約的一部分
- 在選擇了建置系統承包商之後，通常需要經過一段合約協商期間，對變更進行協商

# 承包商與次承包商

- 大型軟硬體系統的採購通常會以一些主要承包商為主
- 然後再與其他供應商簽定子合約，負責某一部分的系統
- 系統客戶與主要承包商溝通，但不直接與次承包商做溝通

# 承包商架構



# 重點整理

- 系統工程是一項複雜且困難的處理程序，需要許多工程專業領域的投入
- 突顯性質是指系統以整體來看時所出現的特性，而不是各個組成元件的性質
- 系統架構通常是以方塊圖來表示，圖中可顯示出各個主要子系統以及它們之間的關係

# 重點整理

- 功能性系統元件的類型包括有感應器元件、觸動元件、運算元件、協調元件、通訊元件以及介面元件等
- 系統工程的程序通常是瀑布式的模型，包括規格制定、設計、開發、整合與測試
- 系統的採購程序包括指定系統、發出要求、選擇供應商，以及承包系統的合約等

# 結論

- 系統工程是不容易的！對複雜系統的開發永遠不會有簡單容易的答案
- 軟體工程並不能得到所有的解答，但是若能以系統的觀點來看會得到較佳的結果
- 不同領域之間必須認清彼此的優勢，並且在系統工程程序中主動合作，而勉強的進行合作