Chapter 9

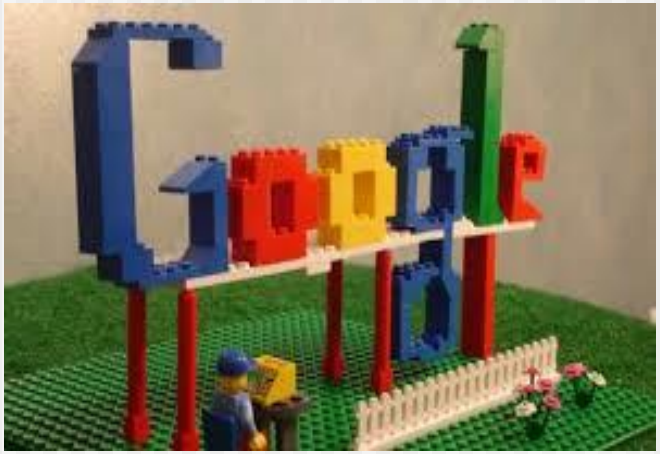
- **Thiết kế kiến trúc**

Slide Set to accompany

Software Engineering: A Practitioner's Approach, 7/e

by Roger S. Pressman

Chapter 9



Why Architecture?

Các kiến trúc không phải là phần mềm hoạt động. Thay vào đó, nó là một đại diện cho phép một kỹ sư phần mềm để:

(1) phân tích hiệu quả của thiết kế trong việc đáp ứng các yêu cầu đề ra,

(2) xem xét lựa chọn thay thế kiến trúc khi thay đổi thiết kế vẫn tương đối dễ dàng, và

(3) giảm thiểu rủi ro gắn liền với việc xây dựng các phần mềm.

Tại sao kiến trúc quan trọng?

- Đại diện của kiến trúc phần mềm là một tạo khả năng for truyền thông giữa tất cả các bên (các bên liên quan) quan tâm đến sự phát triển của một hệ thống dựa trên máy tính.
- Những kiến trúc làm nổi bật thiết kế quyết định ban đầu mà sẽ có một tác động sâu sắc trên tất cả các công việc kỹ thuật phần mềm sau và, quan trọng hơn, vào sự thành công cuối cùng của hệ thống như là một thực thể hoạt động.
- Kiến trúc "tạo thành một, mode m inh bạchtương đối nhỏ về cách hệ thống được cấu trúc và cách các thành phần của nó làm việc cùng nhau" [BAS03].

Mô tả kiến trúc

- IEEE Computer Society đã đề nghị IEEE-Std-1471-2000, *Recommended Practice for Architectural Description of Software-Intensive System*, [IEE00]
  - để thiết lập một khuôn khổ khái niệm và từ vựng cho việc sử dụng trong các thiết kế của kiến trúc phần mềm,
  - để cung cấp hướng dẫn chi tiết cho đại diện cho một mô tả kiến trúc, and
  - khuyến khích thực hành thiết kế kiến trúc âm thanh.
- The IEEE Standard định nghĩa một mô tả kiến trúc (AD) là một "một tập hợp các sản phẩm để tài liệu hoá một kiến trúc.”
  - Các mô tả chính nó được đại diện bằng cách sử dụng nhiều quan điểm, nơi từng xem là "một đại diện của cả một hệ thống từ quan điểm của một tập hợp có liên quan của [các bên liên quan] các mối quan tâm.”

Thể loại kiến trúc

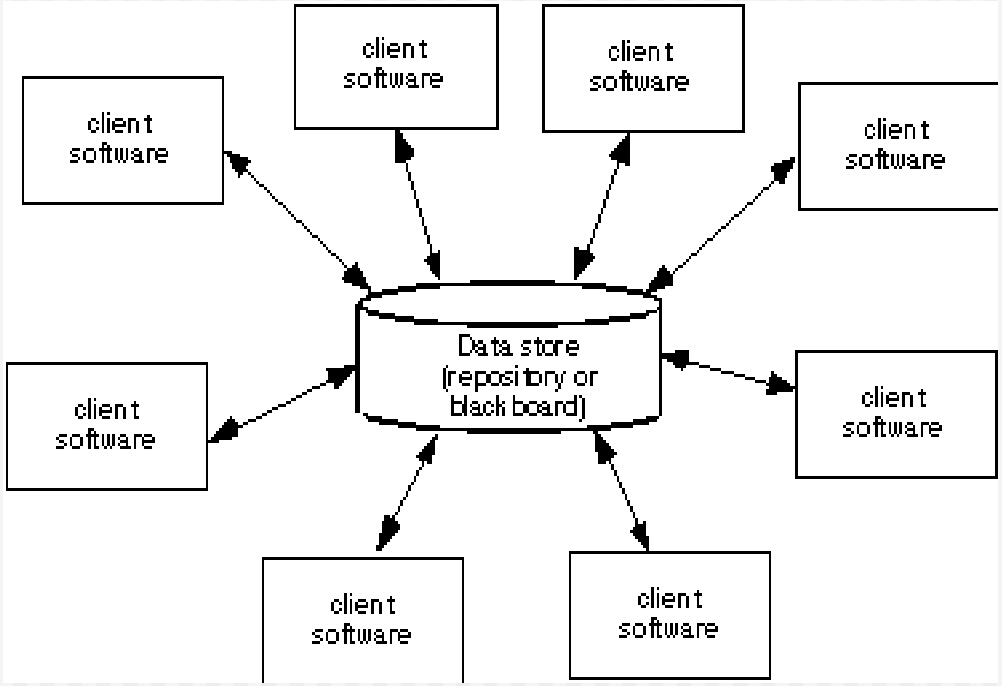
- *Thể loại* ngụ ý một thể loại cụ thể trong lĩnh vực phần mềm tổng thể.
- Trong mỗi thể loại, bạn gặp phải một số tiểu thể loại.
  - Ví dụ, trong các thể loại của các tòa nhà, bạn sẽ gặp phải những phong cách chung sau đây: nhà ở, căn hộ, chung cư, cao ốc văn phòng, tòa nhà công nghiệp, nhà kho, vv.
  - Trong mỗi phong cách chung, phong cách cụ thể hơn có thể được áp dụng. Mỗi phong cách sẽ có một cấu trúc có thể được mô tả bằng một tập các mô hình dự đoán được.

Kiểu kiến trúc

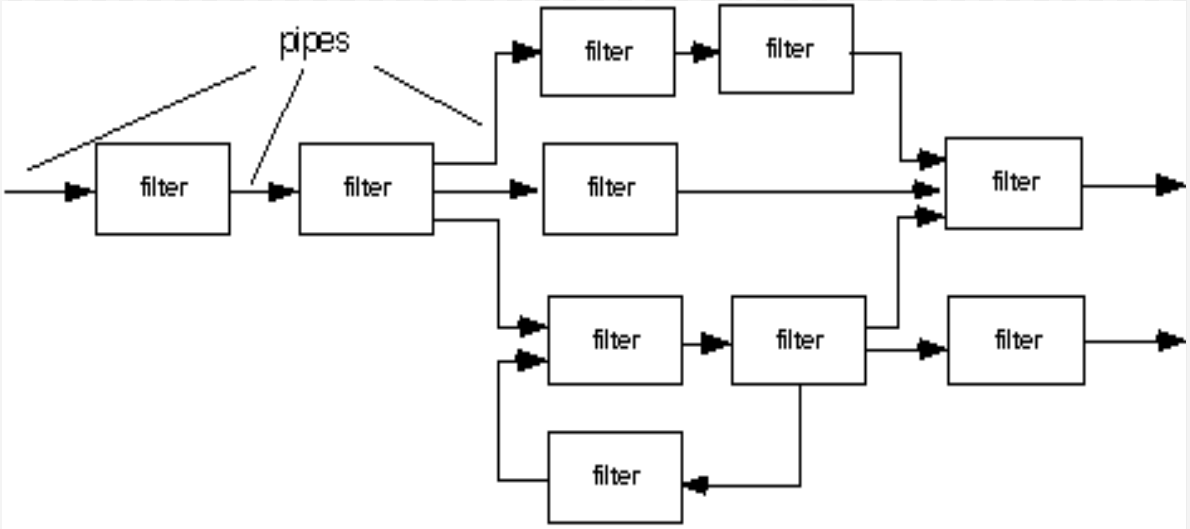
Mỗi phong cách mô tả một loại hệ thống bao gồm: (1) một tập hợp các thành phần (ví dụ, một cơ sở dữ liệu, mô đun tính toán) thực hiện một chức năng cần thiết của một hệ thống, (2) một tập hợp các kết nối cho phép "truyền thông, phối hợp và hợp tác" giữa các thành phần, (3) khó khăn để xác định cách các thành phần có thể được tích hợp để tạo thành hệ thống, và (4) các mô hình ngữ nghĩa cho phép một nhà thiết kế phải hiểu được tính chất tổng thể của hệ thống bằng cách phân tích các đặc tính được biết đến của các bộ phận cấu thành của nó.

- Kiến trúc lấy dữ liệu làm trung tâm
- Kiến trúc luồng dữ liệu
- kiến trúc gọi và trả về
- Kiến trúc hướng đối tượng
- kiến trúc lớp

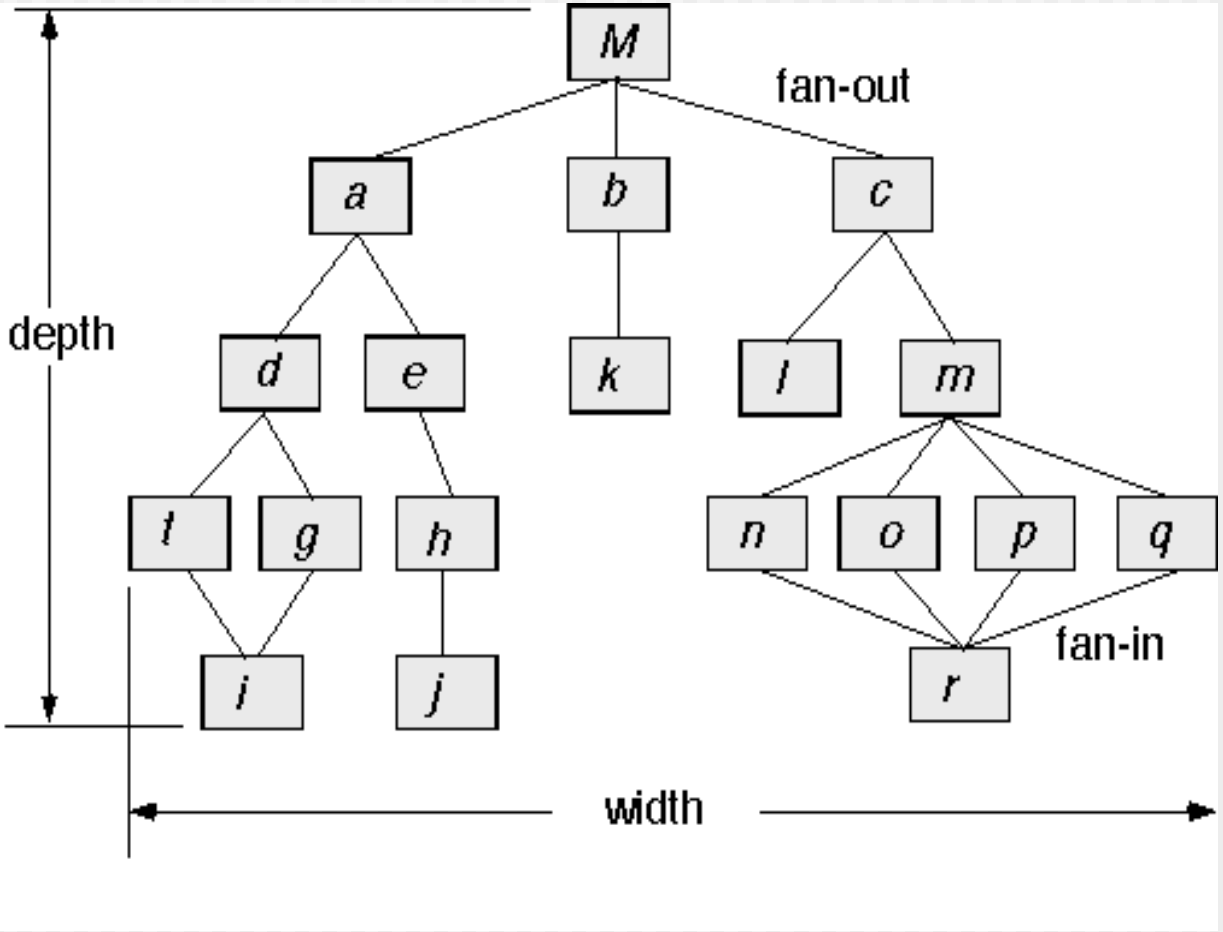
Kiến trúc lấy dữ liệu làm trung tâm



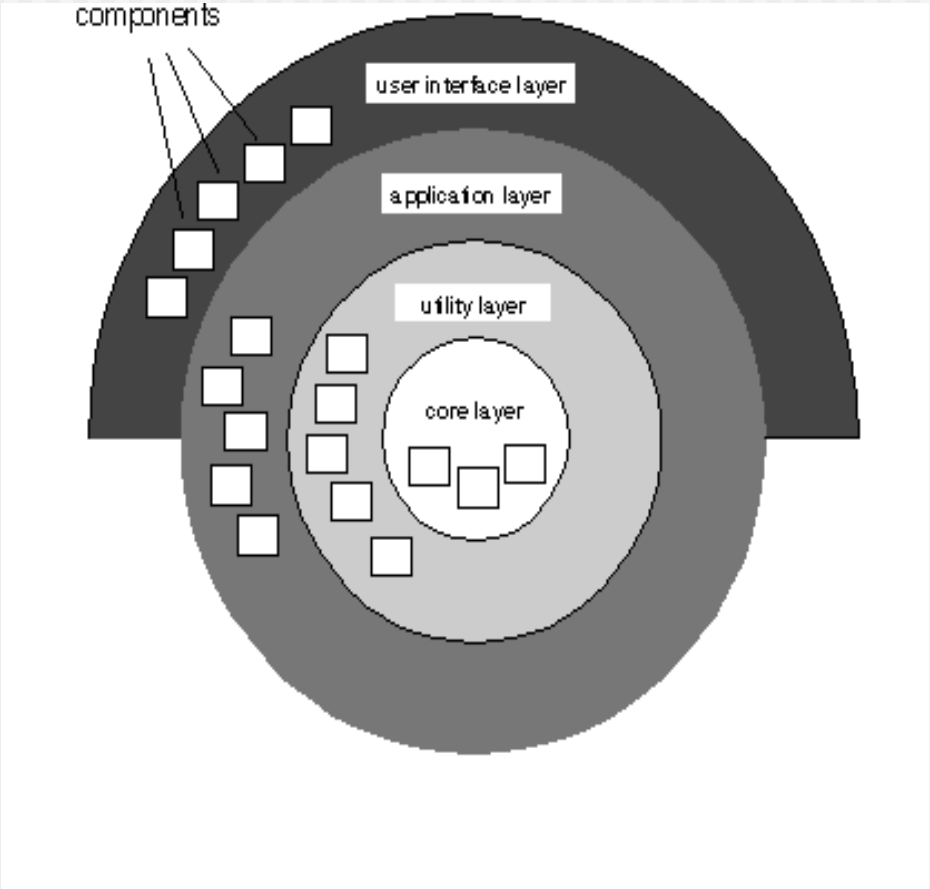
Kiến trúc luồng dữ liệu



kiến trúc gọi và trả về



kiến trúc phân lớp



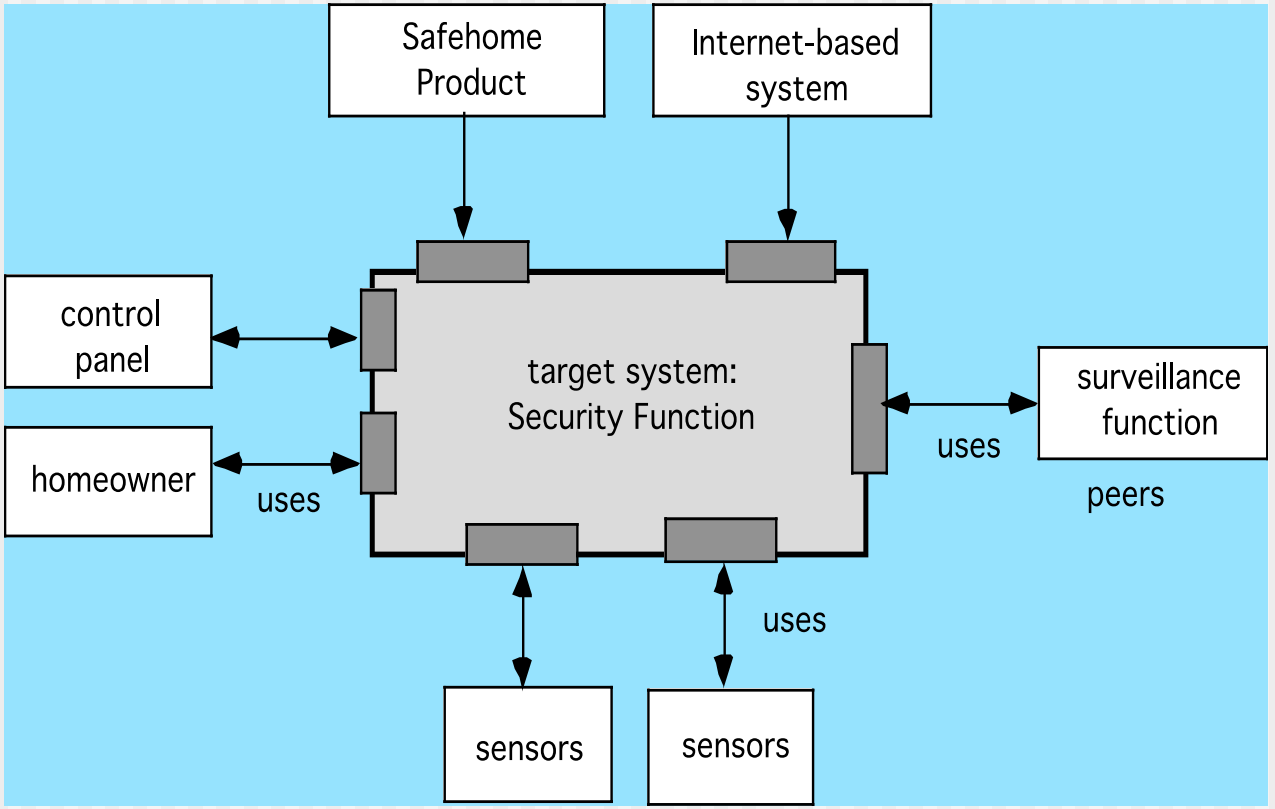
Mô hình kiến trúc

- Đồng thời —các ứng dụng phải xử lý nhiều nhiệm vụ một cách mô phỏng song song
  - *mô hình quản lý điều hành quy trình hệ thống*
  - *bảng kế hoạch* pattern
- Persistence—Dữ liệu vẫn tồn tại nếu nó tồn tại qua việc thực hiện các quá trình tạo ra nó. Hai mô hình phổ biến:
  - một mô hình cơ sở dữ liệu hệ thống quản lý áp dụng lưu trữ và truy xuất khả năng của một DBMS với các kiến trúc ứng dụng
  - một mô hình bền bỉ mức độ ứng dụng mà xây dựng tính năng bền bỉ vào các kiến trúc ứng dụng
- Phân phối— cách thức mà hệ thống hoặc các thành phần trong hệ thống giao tiếp với nhau trong một môi trường phân phối
  - Một nhà môi giới hoạt động như một "người trung gian" giữa các thành phần client và một thành phần máy chủ.

Thiết kế kiến trúc

- Phần mềm này phải được đặt trong bối cảnh
  - thiết kế cần xác định các thực thể bên ngoài (các hệ thống khác, thiết bị, con người) mà phần mềm tương tác với và bản chất của sự tương tác
- Một tập hợp các nguyên mẫu kiến trúc cần được xác định
  - Một nguyên mẫu là một trừu tượng (tương tự như một lớp) đại diện cho một phần tử của hệ thống hành vi
- Các nhà thiết kế xác định cấu trúc của hệ thống bằng cách xác định và tinh chỉnh các thành phần phần mềm thực hiện từng nguyên mẫu

Bối cảnh kiến trúc



Nguyên mẫu

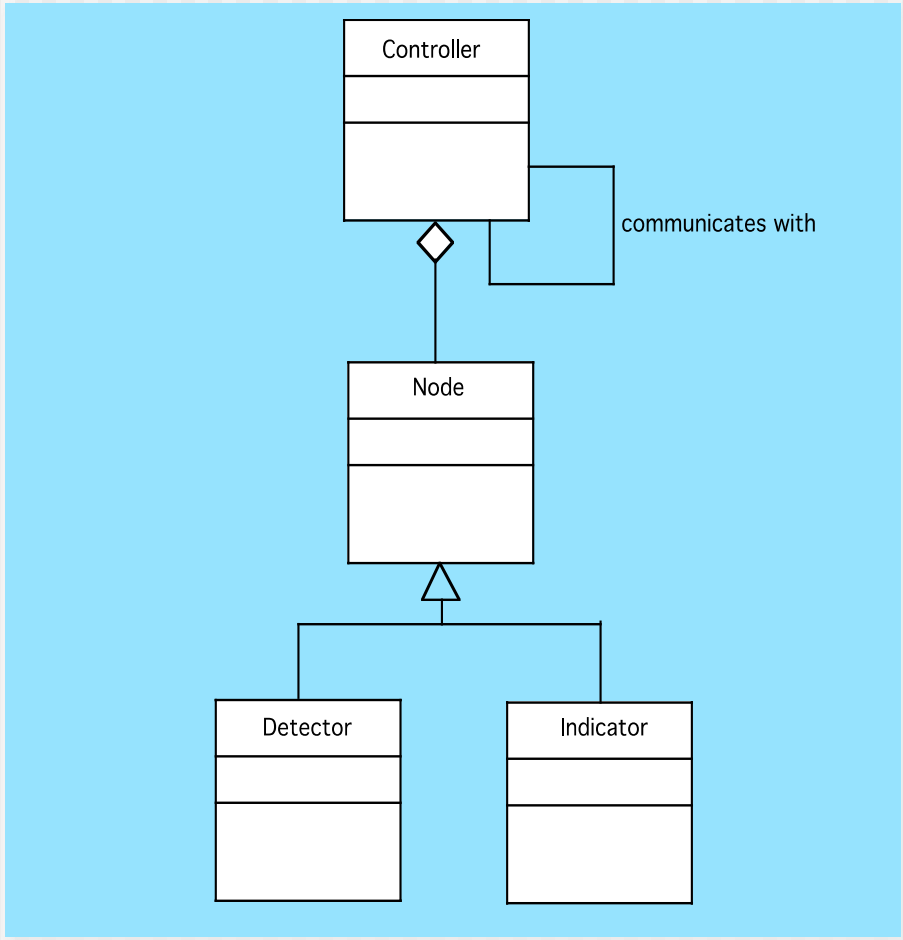
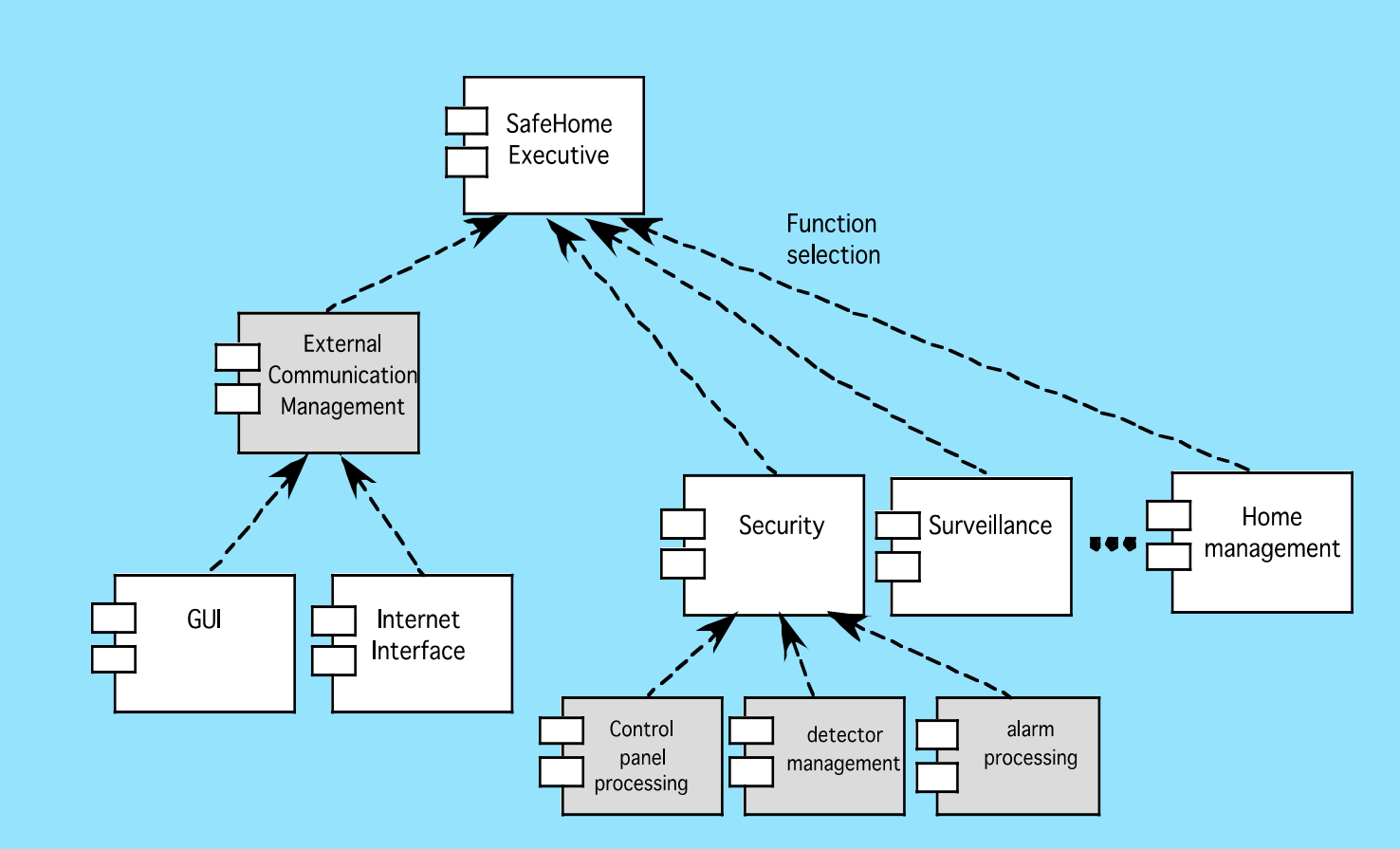
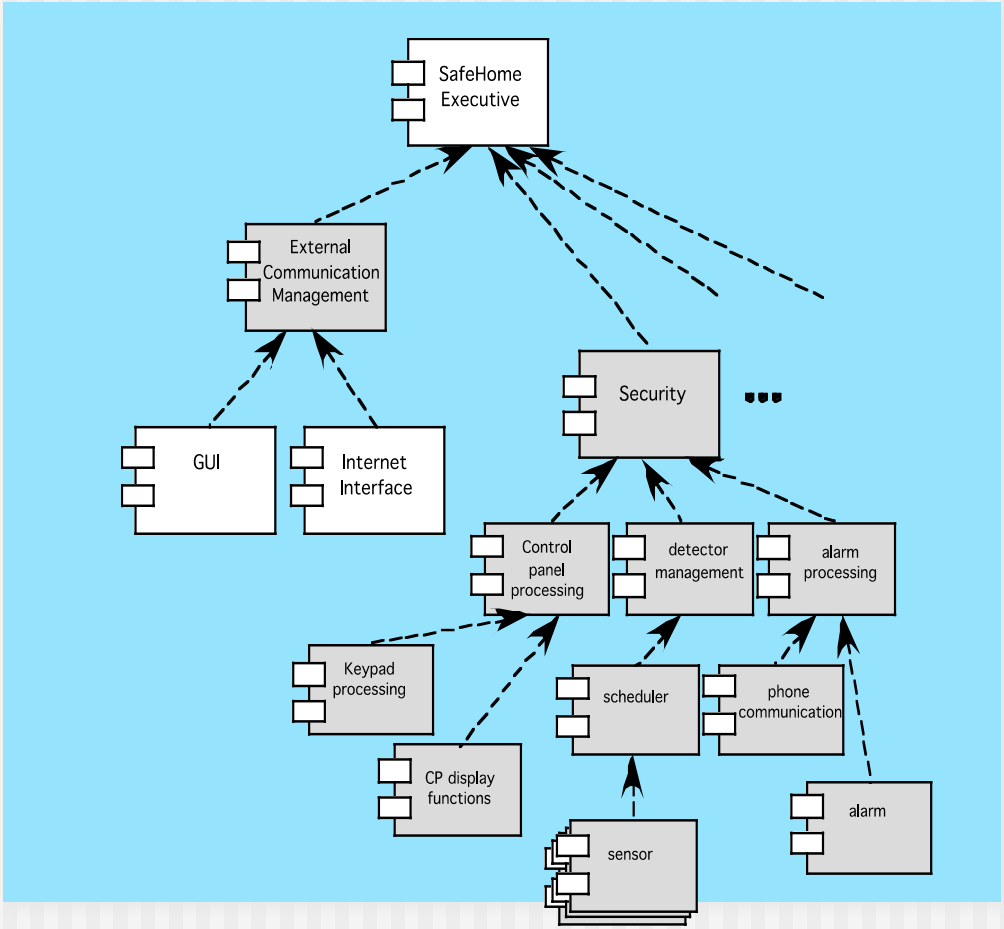


Figure 10.7 UML relationships for Safe Home security function

Cơ cấu thành phần



Cơ cấu thành phần tinh chỉnh



Phân tích thiết kế kiến trúc

1. Thu thập các kịch bản.
2. Gợi ý các yêu cầu, ràng buộc, và đặc tả môi trường.
3. Mô tả các phong cách / mô hình kiến trúc đã được lựa chọn để giải quyết các tình huống và yêu cầu:
    - quan điểm mô-đun
    - góc nhìn quá trình
    - quan điểm dòng dữ liệu
4. Đánh giá chất lượng thuộc tính bằng cách coi mỗi thuộc tính trong sự cô lập.
5. Xác định mức độ nhạy cảm của các thuộc tính chất lượng với các thuộc tính kiến trúc khác nhau cho một phong cách kiến trúc cụ thể.
6. Kiến trúc ứng cử viên phê bình (được phát triển trong bước 3) bằng cách sử dụng phân tích độ nhạy được thực hiện ở bước 5.

Độ phức tạp kiến trúc

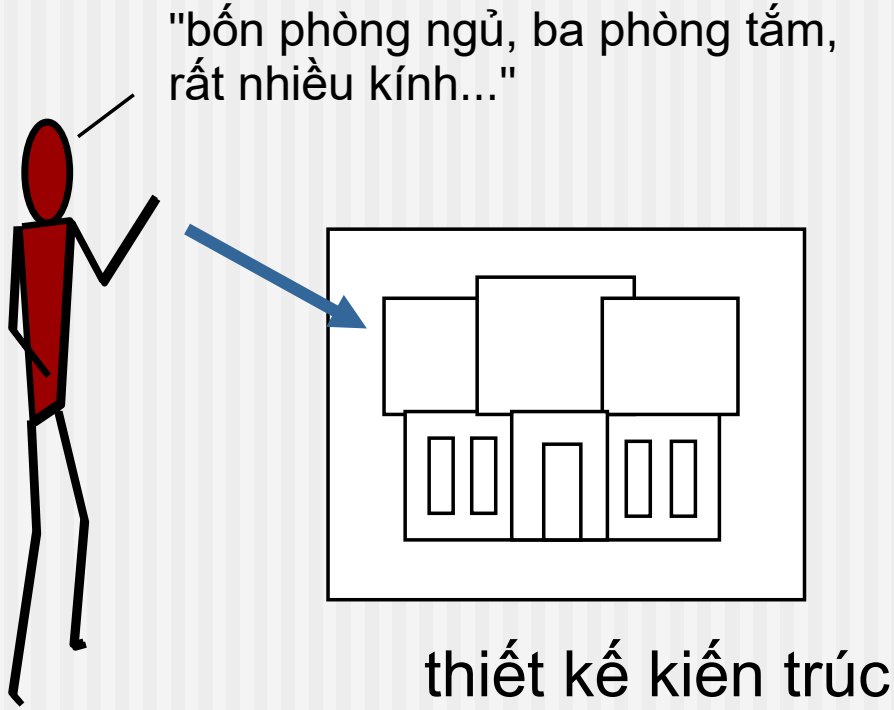
- Độ phức tạp tổng thể của một kiến trúc đề xuất được đánh giá bằng cách xem xét sự phụ thuộc giữa các thành phần trong kiến trúc[Zha98]
  - *Chia sẻ phụ thuộc* đại diện cho mối quan hệ phụ thuộc giữa các khách hàng sử dụng cùng một tài nguyên hoặc các nhà sản xuất đã sản xuất ra cho cùng những người tiêu dùng .
  - *Phụ thuộc luồng* đại diện cho mối quan hệ phụ thuộc giữa sản xuất và tiêu dùng các nguồn lực.
  - *Phụ thuộc ràng buộc* đại diện cho các ràng buộc vào các luồng liên quan của kiểm soát trong một tập hợp các hoạt động.

ADL

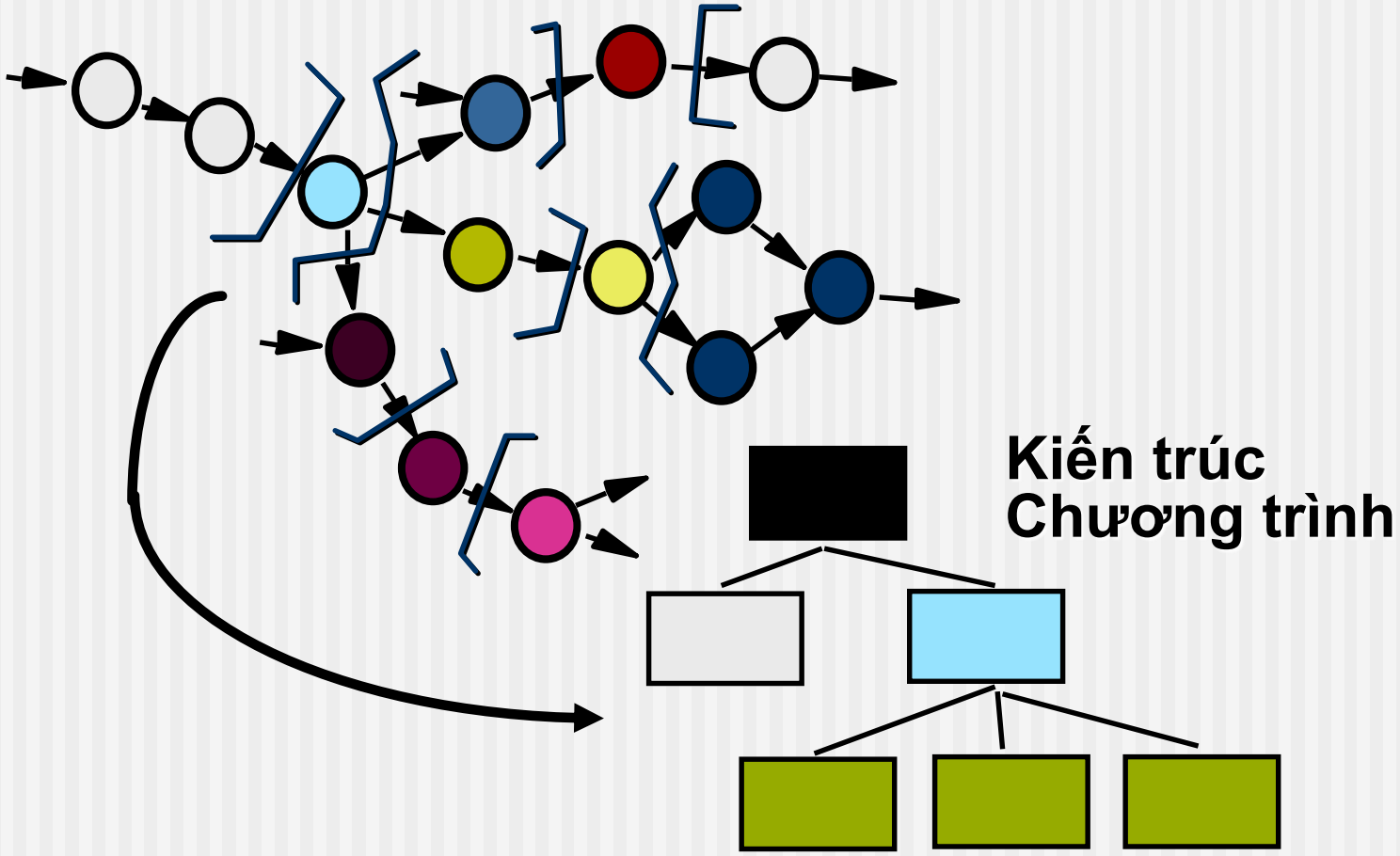
- *Architectural description language* (Ngôn ngữ mô tả kiến trúc-ADL) cung cấp các ngữ nghĩa và cú pháp để mô tả một kiến trúc phần mềm
- Cung cấp cho nhà thiết kế với khả năng:
  - phân giải các thành phần kiến trúc
  - kết hợp thành phần riêng lẻ thành các khối kiến trúc lớn hơn và
  - đại diện cho giao diện (cơ chế kết nối) giữa các thành phần.

Một phương pháp thiết kế kiến trúc

yêu cầu khách hàng

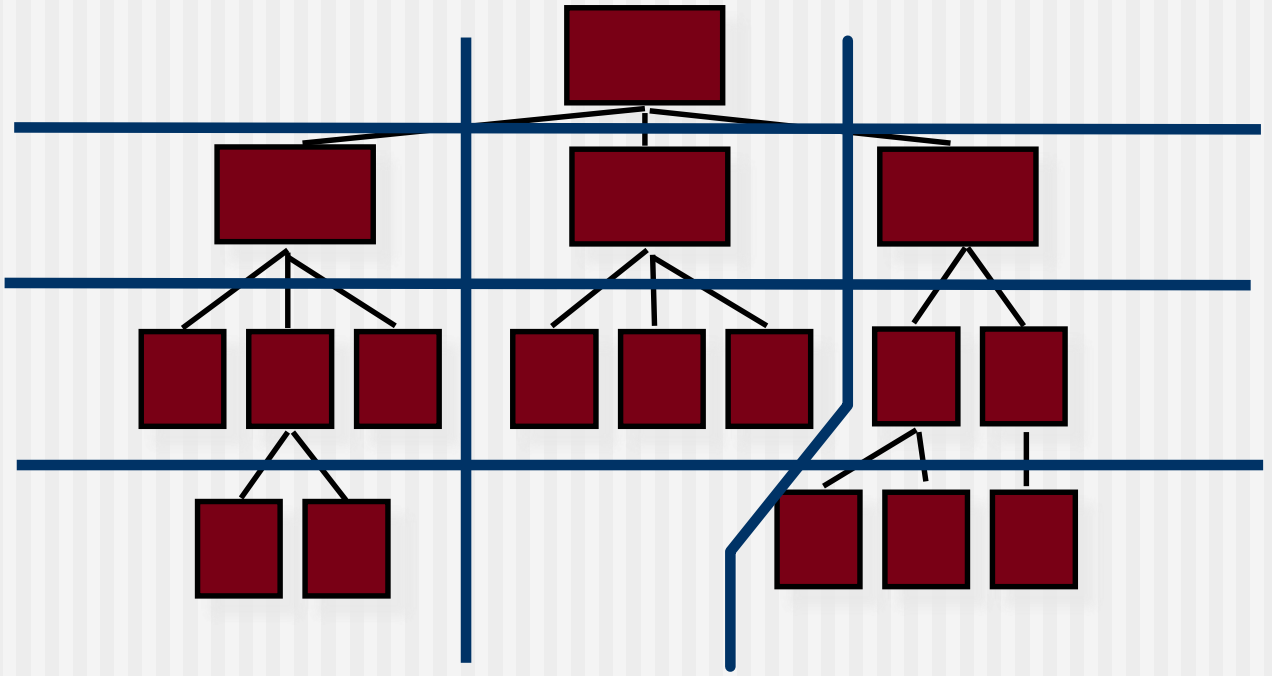


Phát sinh Chương trình Kiến trúc



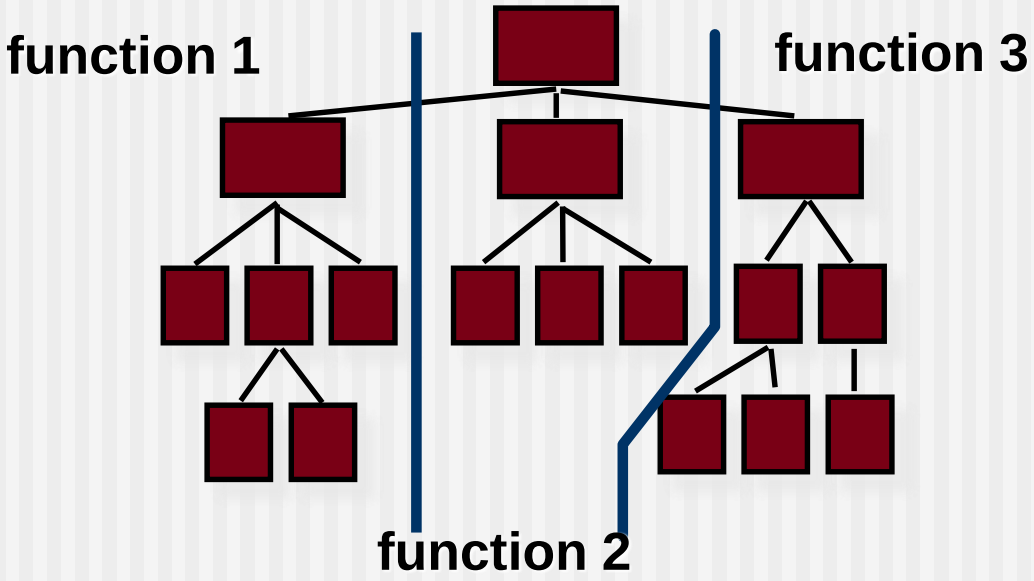
Phân vùng Kiến trúc

- Phân vùng "ngang" và ”dọc" là cần thiết



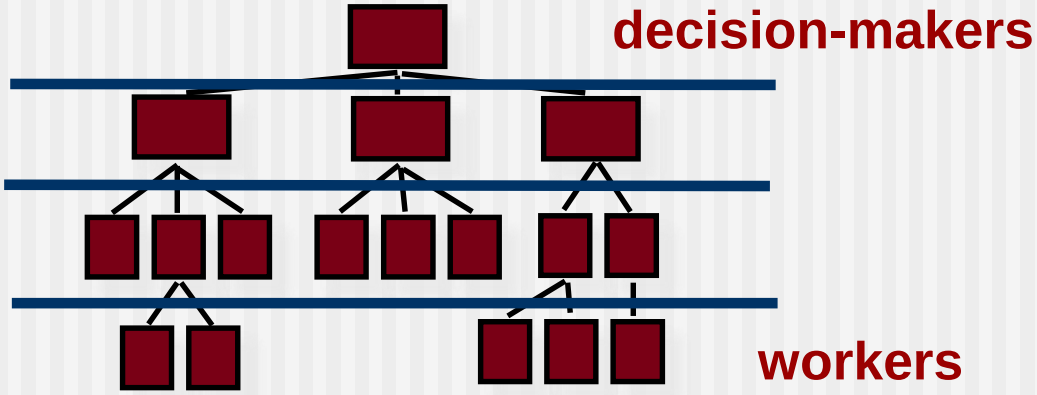
Phân vùng ngang

- xác định các nhánh riêng biệt của hệ thống phân cấp module cho từng chức năng chính
- sử dụng mô-đun điều khiển để phối hợp truyền thông giữa các chức năng



Phân vùng dọc: Factoring

- thiết kế để việc ra quyết định và làm việc được phân tầng
- module ra quyết định nên nằm ở trên cùng của kiến trúc



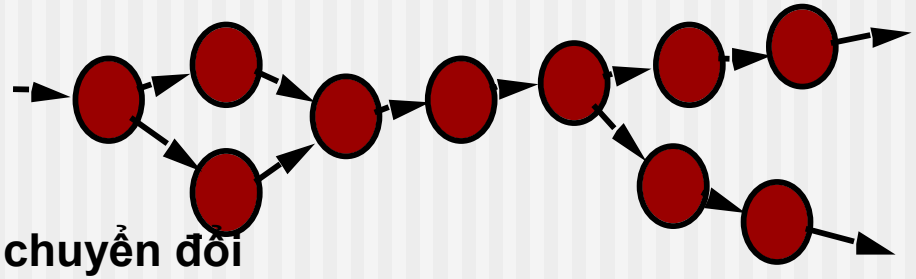
Tại sao Kiến trúc phân vùng?

- kết quả trong phần mềm dễ dàng kiểm tra hơn
- dẫn đến phần mềm dễ dàng để duy trì hơn
- kết quả trong lan truyền ít tác dụng phụ hơn
- kết quả trong phần mềm dễ dàng để mở rộng hơn

Thiết kế cấu trúc

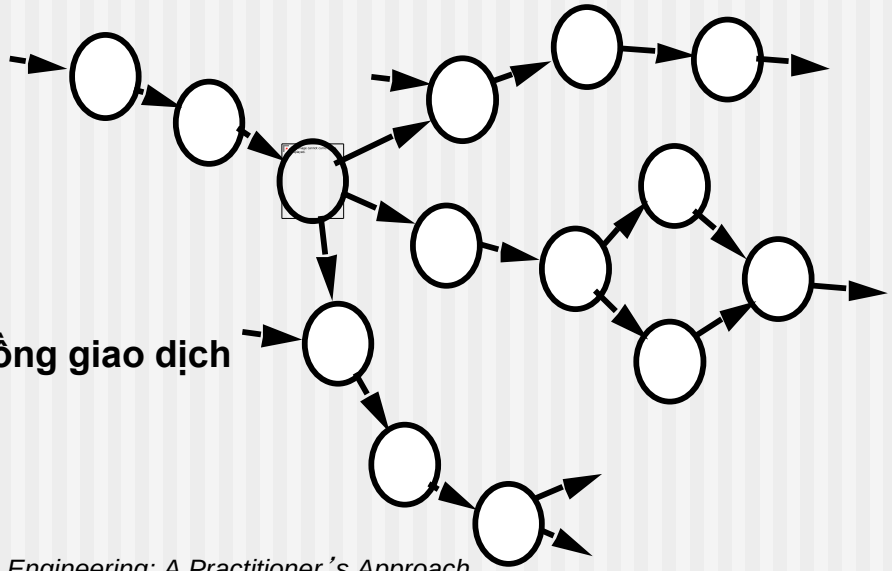
- Mục tiêu: để đưa ra một kiến trúc chương trình được phân chia
- phương pháp tiếp cận:
  - một DFD được ánh xạ vào một kiến trúc chương trình
  - các PSPEC và STD được sử dụng để chỉ ra các nội dung của mỗi mô-đun
- ký pháp: biểu đồ cấu trúc

Các đặc tính luồng



Luồng chuyển đổi

This edition of SEPA does not cover transaction mapping. For a detailed discussion see the SEPA website



Luồng giao dịch

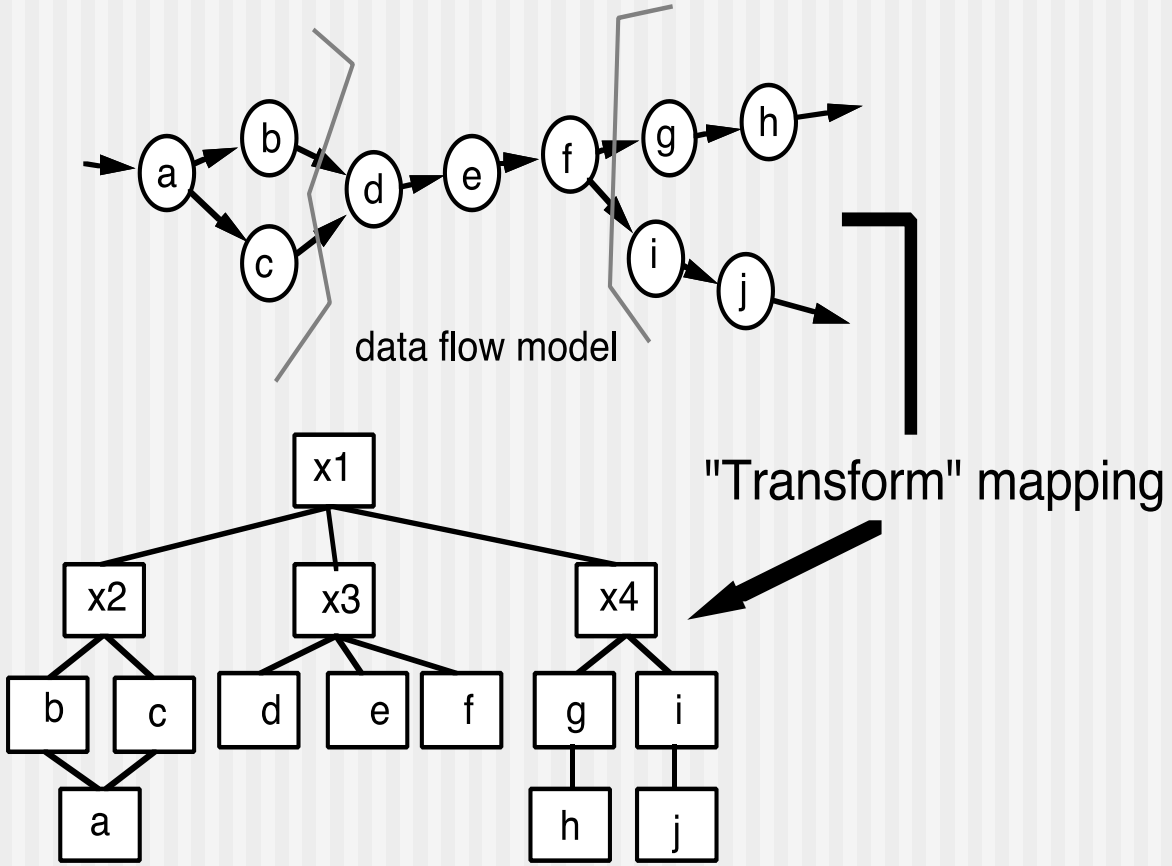
Phương pháp tiếp cận ánh xạ chung

- cô lập ranh giới luồng vào và ra; cho các luồng giao dịch, cô lập các trung tâm giao dịch
- làm việc kể từ ranh giới bên ngoài, ánh xạ DFD biến đổi thành các module tương ứng
- thêm module điều khiển theo yêu cầu
- tinh chỉnh cấu trúc chương trình kết quả bằng cách sử dụng các khái niệm mô đun hiệu quả

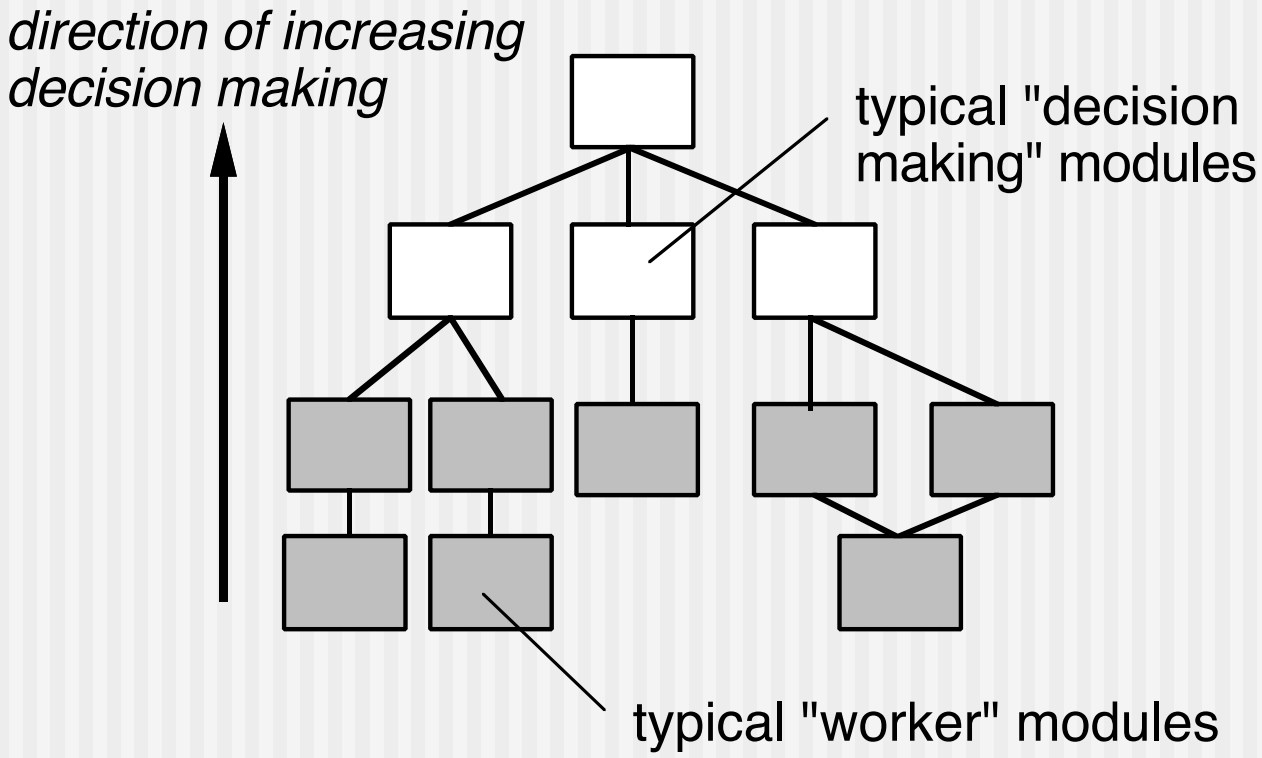
Phương pháp tiếp cận ánh xạ chung

- **Cô lập các trung tâm chuyển đổi bằng cách xác định ranh giới luồng vào và ra**
- **Thực hiện ”factoring.cấp độ đầu tiên”**
  - Các kiến trúc chương trình có nguồn gốc sử dụng kết quả ánh xạ này trong một phân bố trên-xuống của điều khiển.
  - *Factoring* dẫn đến một cấu trúc chương trình trong đó các thành phần cấp cao thực hiện việc ra quyết định và thành phần ở mức độ thấp thực hiện hầu hết công việc đầu vào, tính toán, và đầu ra.
  - Thành phần cấp trung thực hiện một số kiểm soát và làm một lượng vừa phải công việc.
- **Thực hiện ”factoring.cấp độ hai."**

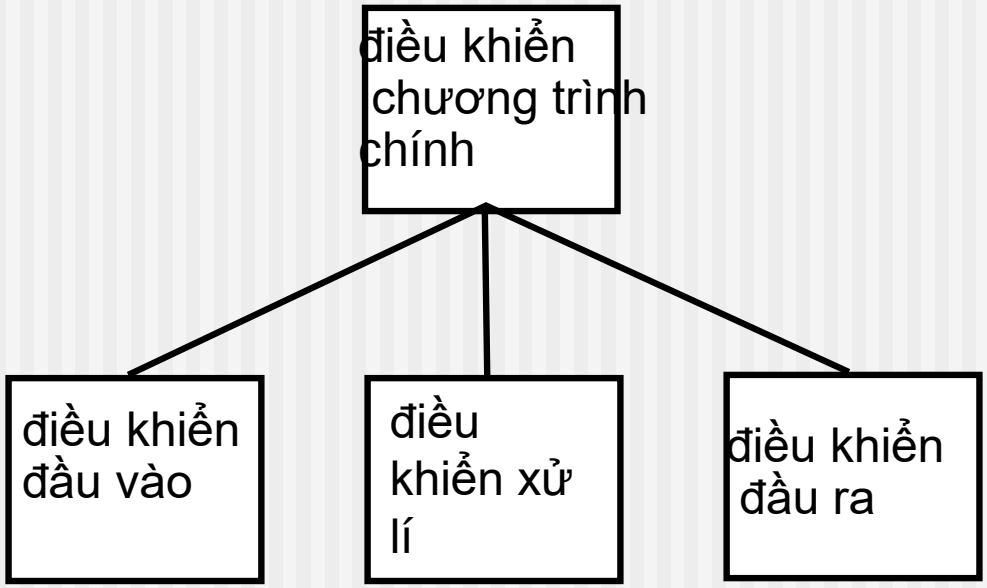
Ánh xạ chuyển đổi



Factoring


direction of increasing
decision making
typical "decision
making" modules
typical "worker" modules

Factoring cấp độ một



Ánh xạ cấp độ hai