



Chương 4.

CÁC KỸ THUẬT TRONG BẢO TRÌ PHẦN MỀM

GVGD: Nguyễn Xuân Hà Giang

- *Đặc điểm thể hiện sự hiểu biết về chương trình*
- *Kỹ thuật đảo ngược*
- *Kỹ thuật tái kiến tạo*

ĐẶC ĐIỂM VỀ SỰ HIỂU BIẾT CHƯƠNG TRÌNH

No.3

CÁC ĐẶC TRƯNG CỦA PM

- *Lĩnh vực của vấn đề*
 - ▣ Chăm sóc sức khỏe, dự báo thời tiết, mô phỏng các hiện tượng thiên nhiên,...
 - ▣ Vấn đề lớn được phân thành bộ phận nhỏ, do 1 hay nhiều đơn vị chương trình khác nhau xử lý
 - ⇒ Trình biên dịch: bộ phân tích từ vựng, bộ phân tích cú pháp, bộ sinh mã trung gian,...
 - ▣ Giúp chọn giải thuật, phương pháp, công cụ
 - ▣ Thông tin lĩnh vực thu được từ hệ thống, end user, mã nguồn,...

ĐẶC ĐIỂM VỀ SỰ HIỂU BIẾT CHƯƠNG TRÌNH

No.4

CÁC ĐẶC TRƯNG CỦA PM

- *Ảnh hưởng từ quá trình thực hiện*
 - ▣ Kiến thức, kinh nghiệm để đánh giá tín hiệu quả của việc thực hiện YCTĐ
 - ▣ Đánh giá mức cao: dự đoán kết quả tổng thể
 - ▣ Đánh giá mức thấp: dự đoán kết quả chi tiết từng thành phần riêng lẻ tạo ra

ĐẶC ĐIỂM VỀ SỰ HIỂU BIẾT CHƯƠNG TRÌNH

No.5

CÁC ĐẶC TRƯNG CỦA PM

□ *Quan hệ nhân quả*

- ▣ Giúp diễn giải cách thức các component tương tác nhau suốt tiến trình
- ▣ Giúp ước lượng phạm vi thay đổi & hiệu ứng dây chuyền
- ▣ Giúp theo vết luồng thông tin suốt chương trình
 - ⇒ Sự gián đoạn luồng thông tin: có thể đó là do lỗi

ĐẶC ĐIỂM VỀ SỰ HIỂU BIẾT CHƯƠNG TRÌNH

No.6

CÁC ĐẶC TRƯNG CỦA PM

- *Quan hệ sản phẩm & môi trường*
 - ▣ Sản phẩm: hệ thống phần mềm
 - ▣ Môi trường: điều kiện ngoài tác động lên sản phẩm (nghịệp vụ, chính sách, mô hình, nền tảng phần cứng, PM)
- ⇒ Dự đoán cách thức môi trường sẽ tác động lên sản phẩm

ĐẶC ĐIỂM VỀ SỰ HIỂU BIẾT CHƯƠNG TRÌNH

No.7

CÁC ĐẶC TRƯNG CỦA PM

- *Các điểm đặc trưng PM để hỗ trợ ra quyết định*
 - Thuộc tính sản phẩm như độ phức tạp, tính BT được,.. giúp BT viên có cơ sở đưa ra quyết định về kỹ thuật & quản lý. Chẳng hạn
 - Đo độ phức tạp: để cân đối tài nguyên
 - Đo tính BT được: để đánh giá chất lượng hệ thống

ĐẶC ĐIỂM VỀ SỰ HIỂU BIẾT CHƯƠNG TRÌNH

No.8

BẢO TRÌ VIÊN VÀ NHU CẦU VỀ THÔNG TIN

- *Quản lý viên: ra quyết định*
 - ▣ Có kiến thức hỗ trợ ra quyết định
 - ▣ Không cần thiết hiểu biết chi tiết thực thi chương trình mức thấp hay kiến trúc hệ thống
 - ▣ Ví dụ:
 - Ước lượng chi phí, thời gian 1 cải tiến lớn cần biết về kích thước PM (dòng lệnh hoặc điểm chức năng)
 - Dùng kết quả ước lượng để quyết định tiếp tục xây dựng hay phát triển hệ thống mới

ĐẶC ĐIỂM VỀ SỰ HIỂU BIẾT CHƯƠNG TRÌNH

No.9

BẢO TRÌ VIÊN VÀ NHU CẦU VỀ THÔNG TIN

- *Phân tích viên: xác định yêu cầu chức năng & phi chức năng*
 - ▣ Thiết lập mối quan hệ hệ thống – thành viên liên quan
 - ▣ Cách thức các thay đổi trong môi trường gây ảnh hưởng hệ thống
 - ▣ Không cần biết chi tiết & cách thức vận hành hệ thống, chi tiết mã lệnh,...
 - ▣ Sử dụng mô hình (ngữ cảnh) để biểu diễn & hỗ trợ phân tích

ĐẶC ĐIỂM VỀ SỰ HIỂU BIẾT CHƯƠNG TRÌNH

No.10

BẢO TRÌ VIÊN VÀ NHU CẦU VỀ THÔNG TIN

- *Thiết kế viên: Mức kiến trúc & mức chi tiết*
 - ▣ Kiến trúc: tạo các component, CSDL mức quan niệm & mối liên kết. Chi tiết: giải thuật, biểu diễn & CTDL, giao diện PM (prototype, routine)
 - ▣ Cung cấp thiết kế & đưa ra cách thức cải tiến
 - ▣ Xem mã nguồn để phát thảo công việc, vùng gây ảnh hưởng hệ thống
 - ▣ Cơ sở hỗ trợ thiết kế: giấu thông tin, phân rã PM, trừu tượng dữ liệu, lưu đồ dòng dữ liệu, điều khiển, biểu đồ cấu trúc, biểu đồ vào ra

ĐẶC ĐIỂM VỀ SỰ HIỂU BIẾT CHƯƠNG TRÌNH

No.11

BẢO TRÌ VIÊN VÀ NHU CẦU VỀ THÔNG TIN

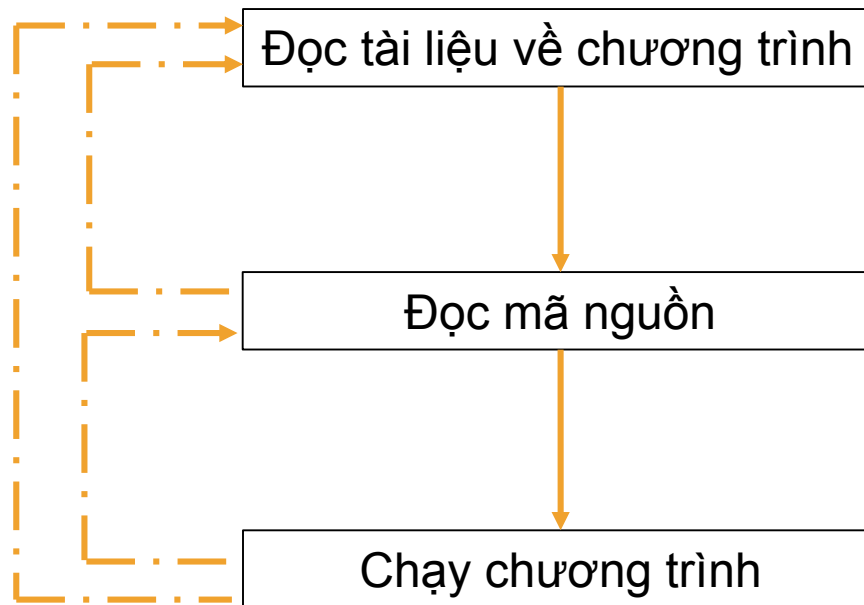
- *Lập trình viên: có kiến thức về*
 - ▣ Ảnh hưởng các mức khác nhau của hệ thống
 - ▣ Quan hệ nhân quả của luồng sự kiện, module
 - ▣ Quan hệ sản phẩm – môi trường
- ⇒ Mức trừu tượng cao (hệ thống): quan hệ giữa các thành phần nghiệp vụ
- ⇒ Mức trừu tượng thấp (module): nhiệm vụ từng khối lệnh, dòng điều khiển, dòng dữ liệu, hàm
- ⇒ *Dự đoán hiệu ứng, lỗi & nguyên nhân, kiến trúc lại khối lệnh....*

ĐẶC ĐIỂM VỀ SỰ HIỂU BIẾT CHƯƠNG TRÌNH

No.12

MÔ HÌNH VỀ VIỆC HIỂU CHƯƠNG TRÌNH

- *Trong thực tế, các công việc trên có thể xuất phát bất kỳ giai đoạn đoạn nào trước*



ĐẶC ĐIỂM VỀ SỰ HIỂU BIẾT CHƯƠNG TRÌNH

No.13

CÁC CHIẾN LƯỢC ĐỂ HIỂU CHƯƠNG TRÌNH

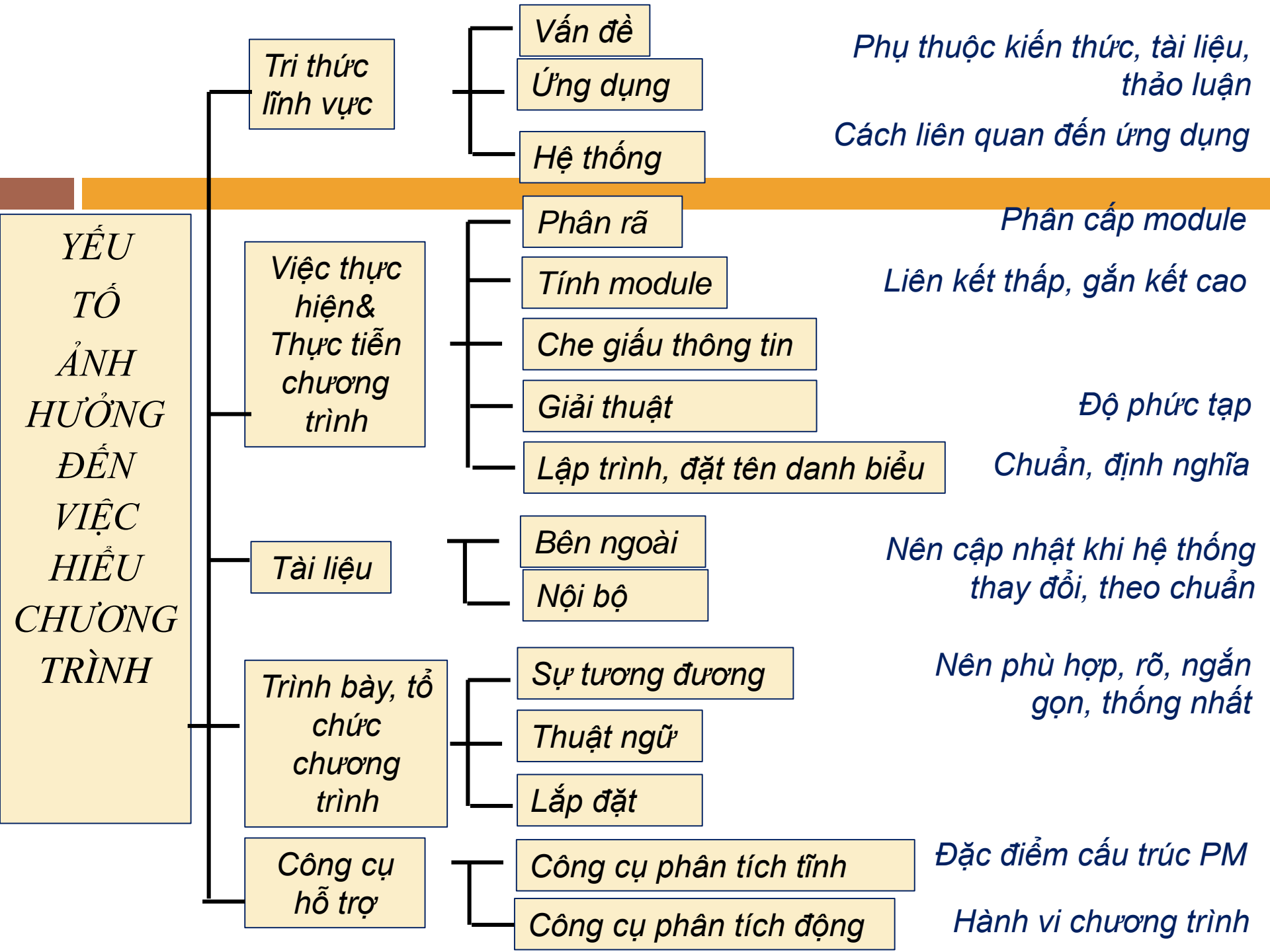
- *Mô hình giúp BT viên hiểu được chương trình*
 - ▣ *Top-down model: trên xuống*
 - ▣ *Bottom-up model: dưới lên, thường ở những BT viên nhiều kinh nghiệm*
 - ▣ *Opportunistic model: cơ hội, kết hợp cả 2 mô hình trên*
- ⇒ *Top-down & bottom-up không tính đến vai trò của công cụ hỗ trợ, hơn nữa thực tế thường vận dụng cả 2 mô hình trên cùng lúc và khi đó thuộc về mô hình cơ hội*

ĐẶC ĐIỂM VỀ SỰ HIỂU BIẾT CHƯƠNG TRÌNH

No.14

CÁC YẾU TỐ ẢNH HƯỞNG ĐẾN VIỆC HIỂU CHƯƠNG TRÌNH

- *Ý kiến chuyên gia: chuyên môn lĩnh vực, ngôn ngữ lập trình giúp hiểu nhanh chương trình, PM*
- *Các vấn đề trong thực hiện: chú thích, độ lòng lênh, rõ, dễ đọc, đơn giản, cấu trúc, lênh, chuẩn....*
- *Tài liệu: thiếu/ đủ, hữu ích/ lỗi thời, đầy đủ/ thất lạc, đúng/ sai...*



- *Quy trình phân tích hệ thống để nhận dạng thành phần (các sản phẩm của quy trình phát triển PM) & mối quan hệ chúng ở mức trừu tượng cao hơn*
- *Giúp hiểu hệ thống làm gì, làm thế nào, kiến trúc ra sao*
- *Điểm bắt đầu đảo ngược phổ biến nhất là code*

- *Đặc trưng, tổng thể về hệ thống bỏ qua chi tiết*
- *Có 3 loại trừu tượng: hàm, dữ liệu & quy trình*
 - ▣ Trừu tượng hàm: suy luận từ đặc trưng quan hệ vào-ra của hàm. Hàm làm gì & không quan tâm làm thế nào
 - ▣ Trừu tượng dữ liệu: suy luận từ dữ liệu & hàm thao tác trên chúng. Lớp dữ liệu & không quan tâm chi tiết thực hiện trên dữ liệu
 - ▣ Trừu tượng quy trình: về thứ tự thực hiện. Có 2 loại: đồng thời & phân tán

- ❑ *Phục hồi thông tin mất*
- ❑ *Di dời giữa các nền tảng*
- ❑ *Cải tiến, cung cấp tài liệu mới*
- ❑ *Rút trích thành phần có thể tái sử dụng*
- ❑ *Giảm công sức BT*
- ❑ *Đổi phó độ phức tạp*
- ❑ *Phát hiện hiệu ứng lề*
- ❑ *Phát triển sản phẩm tương tự & cạnh tranh*

KỸ THUẬT ĐẢO NGƯỢC

No.19 CÁC MỨC KỸ THUẬT ĐẢO NGƯỢC



- *Khi sản phẩm của quy trình đảo ngược*
 - ▣ Cùng mức hệ thống gốc: dẫn chứng lại tài liệu
 - ▣ Mức trừu tượng cao hơn: phục hồi thiết kế hay phục hồi đặc tả
- *Dẫn chứng lại tài liệu: tái tạo lại biểu diễn tương đương ngữ nghĩa*
 - ▣ Cái nhìn khác: giúp hiểu biết. Như: lưu đồ dòng dữ liệu đảo thành lưu đồ dòng điều khiển
 - ▣ Tái tạo lại tài liệu giúp BT tương lai

KỸ THUẬT ĐẢO NGƯỢC

No.21

CÁC MỨC KỸ THUẬT ĐẢO NGƯỢC

- *Phục hồi thiết kế: để đạt mức trù tượng cao hơn, không nhất thiết là thiết kế gốc*
 - ▣ Dùng để tái phát triển hệ thống
 - ▣ Xây dựng hệ thống mới tương tự
 - ▣ Ví dụ: Ứng dụng kiểm tra chính tả để phát triển thành module kiểm tra chính tả
- *Phục hồi đặc tả: khi bản thiết kế không cần thiết để sử dụng lại (do thay đổi về cấu trúc lập trình cấu trúc – hướng đối tượng) hoặc khi đặc tả cũ không còn phù hợp, thất lạc*

KỸ THUẬT ĐẢO NGƯỢC

No.22

LỢI ÍCH KỸ THUẬT ĐẢO NGƯỢC CHO CÁC LOẠI BẢO TRÌ

- Giảm thời gian để hiểu chương trình => giảm chi phí BT
- Khơi nguồn cho phương pháp BT theo nhiều cách
- Hiểu & đánh giá ưu & khuyết điểm các thành phần
- Theo dấu, nhận dạng các biến bị thay đổi và vùng ảnh hưởng
- Hỗ trợ BT thích ứng & hoàn thiện: đơn giản hóa việc hiểu các thành phần chính và quan hệ chúng
- Hỗ trợ BT dự phòng: giúp BT hệ thống tương lai

KỸ THUẬT ĐẢO NGƯỢC

No.23

ĐIỀU KIỆN ĐỀ SỬ DỤNG KỸ THUẬT ĐẢO NGƯỢC

□ Dựa vào chỉ số cảnh báo trong thương mại

Chỉ số	Mô tả	Giải thích
1.	Thiết kế, đặc tả mất, không đủ	Liên quan sản phẩm, môi trường
2.	Tài liệu mất, không chính xác, lạc hậu	
3.	Độ phức tạp chương trình tăng	
4.	Mã nguồn có cấu trúc kém	
5.	Chuyển chương trình sang ngôn ngữ lập trình khác	Liên quan quy trình BT
6.	Tạo ra các sản phẩm tương thích	
7.	Di dời sang nền phần cứng, PM khác	
8.	Lỗi tồn đọng gia tăng	
9.	Năng suất nhân sự giảm	Liên quan thương mại
10.	Cần thay đổi hiệu chỉnh liên tục	
11.	Thay đổi thời gian sống kinh tế của hệ thống	
12.	Cần tạo sản phẩm tương tự nhưng không giống	

- Hạ tầng khi cần đảo ngược mã nguồn (*source code*) sang mã máy (*object code, bytecode*):
 - Hợp ngữ: mỗi nền máy là khác nhau
 - Trình biên dịch: ngôn ngữ cao sang mã máy
 - Object code: theo hệ nền chuẩn cụ thể, có thể thực thi trực tiếp từ CPU
 - Byte code: mã định dạng độc lập với hệ nền
 - Máy ảo, bytecode: máy ảo để chạy bytecode
 - Hệ điều hành: quản lý phần cứng, PM

KỸ THUẬT TÁI KIẾN TẠO

MỤC ĐÍCH

- Giảm sự phức tạp hệ thống
- Giúp PM tiếp tục được dùng, thích nghi, chi phí chấp nhận
- Các thành phần riêng lẻ có thể đưa ra thị trường
- Cải thiện hiệu suất
- Di dời sang hệ nền mới
- Truy xuất và sử dụng để cài đặt hệ nền mới
- Khai thác chuẩn, công nghệ & hệ nền mới

KỸ THUẬT TÁI KIẾN TẠO

CÁC BƯỚC TÁI KIẾN TẠO

□ *Ba giai đoạn*

- Đảo ngược, reversing
- Tái cấu trúc, restructuring
- Chuyển tiếp, forward engineering

Phân tích

Mô hình kiến trúc mức cao

Cải tiến, tái cấu trúc

Mở rộng

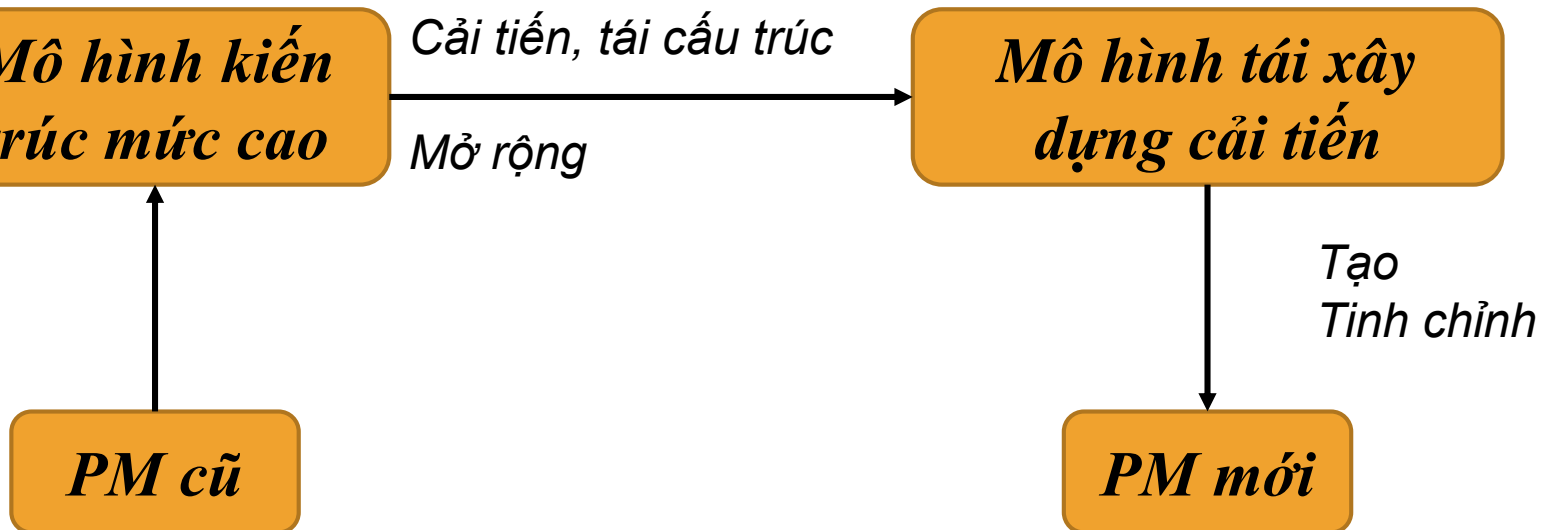
Mô hình tái xây dựng cải tiến

*Hiểu, rút trích
Trừu tượng*

PM cũ

*Tạo
Tinh chỉnh*

PM mới



KỸ THUẬT TÁI KIẾN TẠO

CÁC BƯỚC TÁI KIẾN TẠO

- *Tái cấu trúc: chuyển đổi các biểu diễn trừu tượng (kết quả của kỹ thuật đảo ngược) sang dạng biểu diễn khác cùng mức trừu tượng*
 - ▣ Tái cấu trúc mã nguồn, lược đồ CSDL, không thay đổi hành vi
 - ▣ Phát hiện và loại bỏ mã trùng & mã chết

KỸ THUẬT TÁI KIẾN TẠO

CÁC BƯỚC TÁI KIẾN TẠO

- *Chuyển tiếp: ngược với kỹ thuật đảo ngược*
 - Bắt đầu từ đặc tả, thiết kế, thực thi
 - Hệ thống được tái cấu trúc để cài đặt ở mức trừu tượng thấp
 - Dùng kỹ thuật sinh & sinh tự động source code theo nền hay ngôn ngữ lập trình mới



KỸ THUẬT TÁI KIẾN TẠO

TÁI KIẾN TẠO HỆ THỐNG THÔNG TIN

- *1960s, hệ tập tin*
- *1970s, hệ CSDL. Hiện nay, hệ CSDL gồm phân cấp, mạng, quan hệ, hướng đối tượng & XML*
- *1980s, hệ chuyên gia*
- *Tái kiến tạo thông tin gồm*
 - ▣ *Tái kiến tạo CSDL*
 - ▣ *Tái kiến tạo hệ chuyên gia*

KỸ THUẬT TÁI KIẾN TẠO

TÁI KIẾN TẠO HỆ THỐNG THÔNG TIN

- *Tái kiến tạo CSDL*
 - Dịch lược đồ, schema translation
 - Chuyển đổi dữ liệu
 - Chuyển đổi chương trình
- *Dịch lược đồ: trực tiếp & gián tiếp*
 - Dịch trực tiếp: dịch lược đồ không quan hệ thành lược đồ quan hệ
 - ⇒ Có thể mất thông tin ngữ nghĩa lược đồ gốc
 - ⇒ Dùng dữ liệu nhập NSD nhập để căn cứ phục hồi ngữ nghĩa

KỸ THUẬT TÁI KIẾN TẠO

TÁI KIẾN TẠO HỆ THỐNG THÔNG TIN

- *Dịch lược đồ: trực tiếp & gián tiếp*
 - ▣ Dịch gián tiếp: ánh xạ
 - Lược đồ mạng, lược đồ logic thành lược đồ thực thể- quan hệ khái niệm bằng đảo ngược
 - Lược đồ khái niệm trên tiếp tục được ánh xạ thành lược đồ quan hệ
 - Lược đồ quan hệ được dịch sang mô hình
 - Quan hệ - thực thể, rồi sang UML, sang CSDL đích
 - XML

- *Chuyển đổi dữ liệu*
 - Chuyển vật lý, physical conversion
 - Chuyển logic, logical conversion
 - Chương trình cầu nối, bridge program
- *Chuyển vật lý: dữ liệu vật lý CSDL không quan hệ chuyển sang CSDL quan hệ, bằng*
 - Thông dịch: dịch từng mục
 - Bộ sinh: sinh một chương trình để chuyển đổi dữ liệu vật lý

KỸ THUẬT TÁI KIẾN TẠO

TÁI KIẾN TẠO HỆ THỐNG THÔNG TIN

□ *Chuyển logic:*

⇒ Tập tin trình tự được gỡ bỏ hợp lý khỏi CSDL không quan hệ & tuần tự tái tạo vào CSDL đích

□ *Trong chương trình cầu nối*

⇒ Mỗi tập tin không quan hệ cần chương trình cầu nối để chuyển sang mô hình quan hệ

KỸ THUẬT TÁI KIẾN TẠO

TÁI KIẾN TẠO HỆ THỐNG THÔNG TIN

- *Chuyển đổi chương trình: 5 cách dịch chương trình CSDL không quan hệ sang chương trình CSDL quan hệ*
 - Viết lại: dịch lược đồ và CSDL không quan hệ thành quan hệ, rồi viết lại tất cả chương trình
 - Chương trình cầu nối:
 - Cạnh tranh
 - Dịch ngược
 - Cùng tồn tại

□ Chuyển đổi chương trình:

□ Chương trình cầu nối:

- Chuyển lược đồ không quan hệ ra quan hệ
- Đặt lớp giao diện lên trên hệ quản trị CSDL không quan hệ
- Giao diện dịch DML (data-manipulation language – thao tác dữ liệu) của chương trình quan hệ thành lệnh DML không quan hệ để truy xuất CSDL không quan hệ

KỸ THUẬT TÁI KIẾN TẠO

TÁI KIẾN TẠO HỆ THỐNG THÔNG TIN

- *Chuyển đổi chương trình:*
 - Cạnh tranh:
 - Dùng PM ánh xạ lệnh chương trình nguồn vào lệnh tương đương chương trình đích
 - Cùng tồn tại: tiếp tục hỗ trợ CSDL không quan hệ hiện tại trong khi phát triển CSDL quan hệ tương đương cùng ứng dụng

KỸ THUẬT TÁI KIẾN TẠO

TÁI KIẾN TẠO HỆ THỐNG THÔNG TIN

- *Tái kiến tạo hệ chuyên gia, hệ CSDL*
 - ▣ Hệ cơ sở tri thức và hệ CSDL trong hệ chuyên gia xây dựng trên nhiều môi trường và dự án khác nhau
 - ▣ Có 2 nhóm tái kiến tạo của loại này
 - Nhóm 1: cải tiến hệ CSDL, tri thức hiện tại, kết nối client-server, peer-to-peer giữa CSDL & hệ chuyên gia
 - Nhóm 2: tìm kiếm mô hình biểu diễn tri thức mới

KỸ THUẬT TÁI KIẾN TẠO

TÁI KIẾN TẠO HƯỚNG ĐỐI TƯỢNG

- *Kiến tạo PM phát triển theo lối truyền thống thành hướng đối tượng. Các bước thực hiện*
 - ▣ Đảo ngược PM hiện tại, sinh ra mô hình dữ liệu, hành vi, chức năng
 - ▣ Tạo usecase khi mở rộng chức năng, hành vi
 - ▣ Dùng mô hình dữ liệu, lớp, trách nhiệm, cộng tác để định nghĩa lớp
 - ▣ Xây dựng hệ thống con theo định nghĩa và thiết kế hướng đối tượng

KỸ THUẬT TÁI KIẾN TẠO

TÁI KIẾN TẠO HƯỚNG ĐỐI TƯỢNG

- *Kiến tạo ngay chính PM hướng đối tượng*
 - Tài liệu chưa hoàn chỉnh
 - Phân lớp không đúng
 - Tính module hóa thiếu (liên kết module cao làm giảm độ khả chuyển)
 - Mã trùng, mã chết
 - Chứng năng trùng do nhiều nhóm thiết kế riêng biệt

KỸ THUẬT TÁI KIẾN TẠO

TÁI KIẾN TẠO GIAO DIỆN NGƯỜI DÙNG

- *Dùng đảo ngược để trừu tượng hóa mô hình khái niệm giao diện NSD & dùng chuyển tiếp để tái cài đặt bằng công nghệ khác*
- *Phương pháp của Merlo*
 - ▣ *Hiểu giao diện, luồng dữ liệu trong ứng dụng*
 - ▣ *Tổ chức lại hành vi (giao diện cũ) thành chuỗi có nghĩa trong ngữ cảnh mới*
 - ▣ *Đưa ra cải tiến chế độ tương tác hiệu quả hơn*
 - ▣ *Xây dựng, tích hợp giao diện mới*

KỸ THUẬT TÁI KIẾN TẠO

TÁI KIẾN TẠO GIAO DIỆN NGƯỜI DÙNG

□ *Phương pháp của Morph*

- Tái kiến tạo giao diện ký tự sang đồ họa
- Tái kiến tạo giao diện đồ họa sang hệ nền mới
- Gồm 3 bước
 - Phát hiện: phân tích mã nguồn để trích xuất mô hình giao diện NSD
 - Biểu diễn: xây dựng mô hình trừu tượng phân lớp, lá: giao diện, chức năng: nút trong
 - Chuyển đổi: tập quy tắc chuyển đổi mô hình thành giao diện

KỸ THUẬT TÁI KIẾN TẠO

ĐIỀU KIỆN ĐỂ SỬ DỤNG KỸ THUẬT TÁI KIẾN TẠO

- *Các dấu hiệu để cần dùng kỹ thuật tái kiến tạo*
 - ▣ Tài liệu lỗi thời, không có, hiệu hạn chế
 - ▣ Thiếu kiểm thử
 - ▣ Người phát triển không còn làm việc
 - ▣ Tài liệu và source code không đồng bộ
 - ▣ Tốn nhiều giờ để thực hiện thay đổi đơn giản
 - ▣ Cần sửa lỗi định kỳ
 - ▣ Khó tách phiên bản cho khách hàng khác nhau
 - ▣ Mã lệnh trùng lặp, rối, dài, không hiệu quả

- Câu 1: Tại sao trong BTPM, hiểu chương trình là quan trọng?
- Câu 2: Yếu tố nào ảnh hưởng đến việc hiểu chương trình & yếu tố nào quan trọng nhất?
- Câu 3: Giải thích sự khác nhau giữa các mức kỹ thuật đảo ngược và ví dụ?
- Câu 4: Nêu 1 sản phẩm PM nhận BT có sử dụng kỹ thuật đảo ngược, nếu những lý do vì sao sử dụng kỹ thuật này

- *Câu 5: Giải thích sự khác nhau giữa kỹ thuật đảo ngược & kỹ thuật chuyển tiếp*
- *Câu 6: Ngân hàng lớn dùng PM khoảng 1 triệu dòng lệnh, viết bằng Cobol, vận hành 20 năm, để quản lý tài khoản và vay nợ. Nhiều năm sửa đổi, hệ thống quá tốn kém nếu BT. Đưa lời khuyên?*
- *Giới thiệu 1 PM cần tái kiến tạo? Lý do sử dụng và cách thực hiện?*