



KIẾN TRÚC PHẦN MỀM HƯỚNG THÀNH PHẦN (COMPONENT BASED ARC)

Giảng viên: Nguyễn Thu Phương
Email: ntphuong@ictu.edu.vn

Nội dung



❖ Mục tiêu của bài này là giới thiệu các kiểu kiến trúc hướng thành phần:

- Giới thiệu về kiến trúc
- Ưu điểm
- Nhược điểm
- Ứng dụng



Kiến trúc hướng thành phần

- ❖ Kiến trúc thành phần dựa trên mô tả một phương pháp công nghệ phần mềm để thiết kế và phát triển hệ thống.
- ❖ Nó tập trung vào việc phân rã các thiết kế thành các thành phần chức năng hoặc các thành phần logic có chứa các phương pháp, các sự kiện, và thuộc tính.



Kiến trúc hướng thành phần

❖ What is a component?

- Một thành phần là một phần của phần mềm hoặc thiết kế phần mềm với “giao diện rõ ràng và nội dung bên trong bị ẩn”.
- Giao diện của thành phần chứa các tính năng của nó cho thấy những gì các thành phần có thể cung cấp. Ví dụ các tính năng như là các dịch vụ, các sự kiện, các thuộc tính, các công cụ.

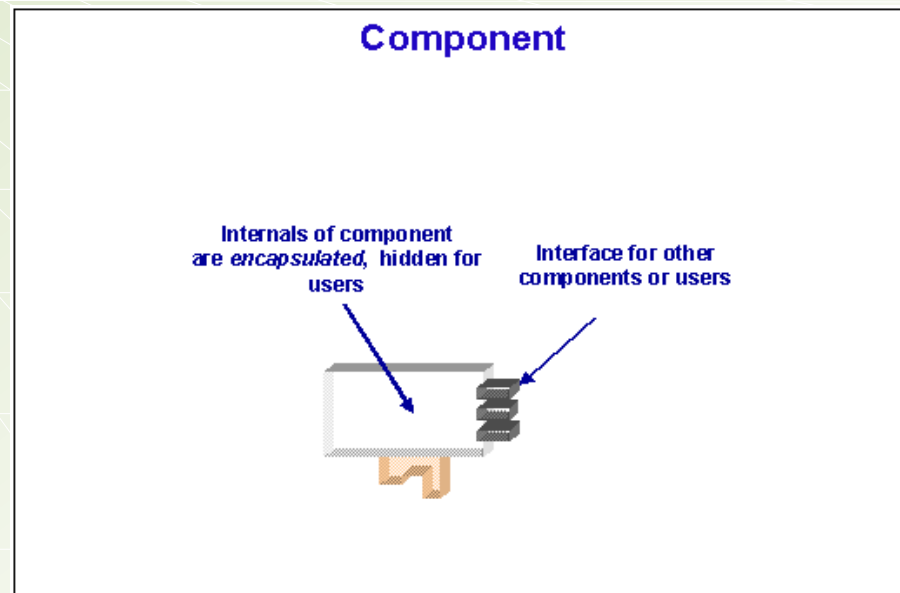


Figure 3.2





Kiến trúc hướng thành phần

❖ Xây dựng ứng dụng dựa trên thành phần:

- Mỗi thành phần đảm nhiệm một phần công việc nhất định.
- Các thành phần này phải tuân theo tiêu chuẩn nhất định được xác định trước bao gồm giao diện, các kết nối, phiên bản, và triển khai.
- Về nguyên tắc, mỗi thành phần có thể tái sử dụng độc lập với ngữ cảnh, nói cách khác có thể dễ dàng sử dụng tại mọi lúc, mọi nơi.
- Sau đó, người ta ghép các thành phần với nhau để thu được một phần mềm thoả mãn một tập các yêu cầu xác định.



Kiến trúc hướng thành phần

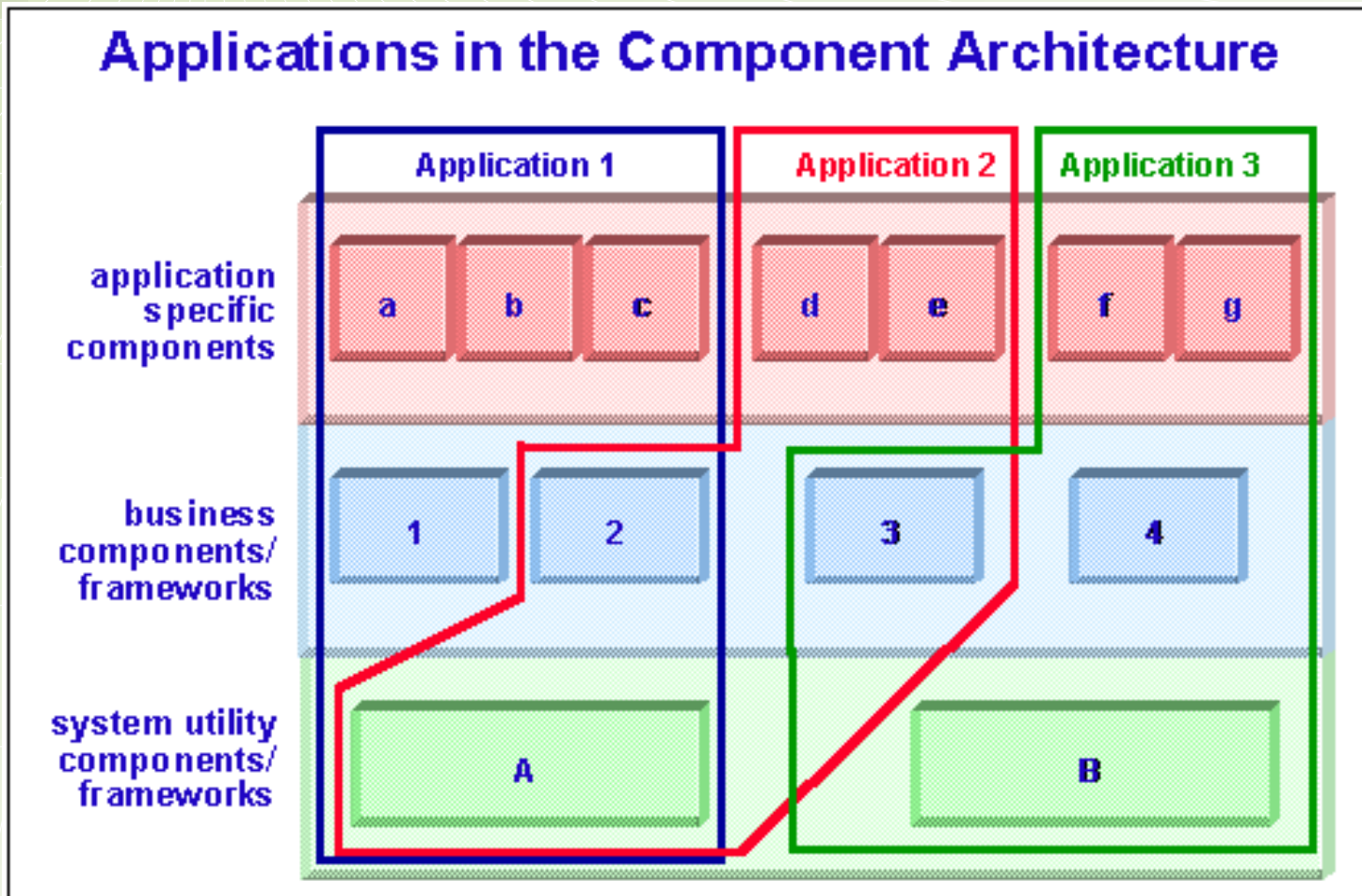
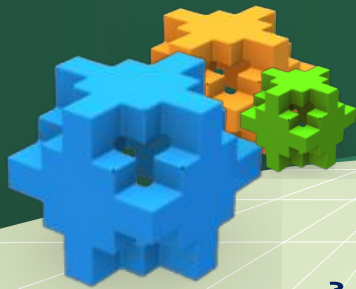


Figure 8.6





Kiến trúc hướng thành phần

❖ Đặc điểm của kiến trúc hướng thành phần:

- Tái sử dụng: Thành phần thường được thiết kế để được tái sử dụng trong các tình huống khác nhau trong các ứng dụng khác nhau. Tuy nhiên, một số thành phần có thể được thiết kế cho một công việc cụ thể.
- Có khả năng thay thế: Các thành phần có thể được dễ dàng thay thế bằng các thành phần tương tự khác.
- Không bối cảnh cụ thể: Các thành phần được thiết kế để hoạt động trong các môi trường và bối cảnh khác nhau.



Kiến trúc hướng thành phần

❖ Đặc điểm của kiến trúc hướng thành phần:

- Có khả năng mở rộng: Một thành phần có thể được mở rộng từ các thành phần hiện có bằng việc cung cấp cho hành vi mới.
- Đóng gói
- Độc lập: Các thành phần được thiết kế để có phụ thuộc tối thiểu trên các thành phần khác. Vì vậy các thành phần có thể được triển khai vào bất kỳ môi trường thích hợp mà không ảnh hưởng đến các thành phần khác hoặc các hệ thống khác.



Kiến trúc hướng thành phần

❖ Ưu điểm của kiến trúc này:

- Dễ dàng triển khai: Như các phiên bản tương thích mới trở nên có sẵn, bạn có thể thay thế phiên bản hiện tại không có tác động trên các thành phần khác hoặc toàn bộ hệ thống.
- Chi phí giảm. Việc sử dụng các thành phần của bên thứ ba cho phép bạn giảm các chi phí phát triển và bảo trì.
- Dễ dàng phát triển. Các thành phần thực hiện các giao diện rõ ràng để cung cấp chức năng xác định, cho phép phát triển mà không ảnh hưởng đến các bộ phận khác của hệ thống.



Kiến trúc hướng thành phần

❖ Ưu điểm của kiến trúc này:

- Tái sử dụng. Việc sử dụng các thành phần tái sử dụng có nghĩa là chúng có thể được sử dụng để phát triển cho các hệ thống khác.
- Tăng năng suất phần mềm: Phần mềm được phát triển dựa trên cấu phần được xây dựng bằng cách lắp ráp các cấu phần tái sử dụng hiện tại. Quá trình này nhanh hơn nhiều so với phát triển một ứng dụng từ vạch xuất phát.
- Tăng độ tin cậy của phần mềm: các cấu phần có thể tái sử dụng thường được thử nghiệm thông qua quá trình xác nhận và sử dụng thực sự trong một thời gian dài, và do đó chất lượng của chúng luôn được đảm bảo.



Kiến trúc hướng thành phần

- ❖ Một số ứng dụng của kiến trúc hướng thành phần:
 - Trong phần cứng: trong các công nghệ lắp ráp
 - Trong phần mềm: sử dụng trong các hệ thống web, windows applications.

Câu hỏi?