

Review sách chapter 18: Emerging Technologies

I. Nội dung:

1. Aspect-Oriented Technology(AOT):

Tập trung vào việc tách biệt các mối quan tâm (concerns), đặc biệt là các mối quan tâm cắt ngang (cross-cutting concerns) như ghi nhật ký kiểm toán (audit trail). Aspect-Oriented Programming (AOP) sử dụng các khái niệm như advice (mã thực thi) và pointcut (vị trí áp dụng mã) để cô lập các mối quan tâm này, giảm lỗi hồi quy (regression faults) khi bảo trì. Ví dụ: AspectJ là ngôn ngữ AOP phổ biến.

2. Model-Driven Technology (MDT):

Model-Driven Architecture (MDA) tách biệt chức năng phần mềm khỏi nền tảng triển khai thông qua mô hình độc lập nền tảng (PIM) và mô hình đặc thù nền tảng (PSM). PIM được xây dựng bằng UML hoặc ngôn ngữ đặc thù, sau đó chuyển đổi thành PSM và mã nguồn tự động, tăng tính di động (portability).

3. Component-Based Technology (CBT):

Nhằm xây dựng phần mềm từ các thành phần tái sử dụng, độc lập, được đóng gói hoàn toàn (encapsulated). CBT hứa hẹn tăng năng suất và chất lượng, giảm thời gian phát triển, nhưng vẫn đối mặt với thách thức về chuẩn hóa và truy xuất thành phần.

4. Service-Oriented Technology (SOT):

Cung cấp khả năng thông qua dịch vụ mạng (ví dụ: Google Docs), nơi người dùng không cần cài đặt phần mềm. SOT nhấn mạnh tính tái sử dụng, đóng gói, và khả năng truy cập từ xa.

5. So sánh SOT và CBT:

Cả hai đều là công nghệ phân tán, nhấn mạnh tái sử dụng, đóng gói, và giao tiếp qua giao diện. Tuy nhiên, SOT sử dụng chương trình thực thi hoàn chỉnh, trong khi CBT xây dựng từ các thành phần. SOT đã được áp dụng rộng rãi, còn CBT cần nghiên cứu đột phá.

6. Social Computing:

Đề cập đến hai khía cạnh: (1) hỗ trợ hành vi xã hội (như mạng xã hội, wiki); (2) tính toán nhóm (như đấu giá trực tuyến, game nhiều người chơi). Công nghệ này tập trung vào tương tác xã hội hoặc xử lý dữ liệu lớn.

7. Web Engineering:

Một nhánh của kỹ thuật phần mềm, tập trung vào phát triển phần mềm web không lỗi, đúng hạn, đúng ngân sách. Phần mềm web có đặc điểm như yêu cầu không ổn định, người dùng đa dạng, và thời gian bảo trì ngắn.

8. Cloud Technology:

Đồng nghĩa với công nghệ dựa trên Internet, nơi người dùng không cần biết hạ tầng bên dưới. Ví dụ: lưu trữ và xử lý dữ liệu trên đám mây.

9. Web 3.0 (Semantic Web):

Dự đoán cách Web sẽ được sử dụng trong tương lai, với nhiều đề xuất nhưng chưa rõ ràng về kết quả cụ thể.

10. Computer Security:

Không phải nhánh của kỹ thuật phần mềm, nhưng liên quan chặt chẽ. Vấn đề “dancing pigs” chỉ ra rằng người dùng ưu tiên tính năng hơn bảo mật, đòi hỏi tích hợp bảo mật vào phần mềm từ đầu.

11. Model Checking:

Công nghệ kiểm tra tính đúng đắn của phần mềm bằng cách mô hình hóa phần mềm dưới dạng máy trạng thái hữu hạn (finite state machine) và sử dụng logic thời gian (temporal logic), thay thế chứng minh thủ công.

II. Bài học:

- Tách biệt mối quan tâm tăng hiệu quả bảo trì
- Tăng tính di động qua mô hình hóa
- Tái sử dụng là chìa khóa nâng cao năng suất
- Hiểu rõ sự khác biệt giữa CBT và SOT
- Web Engineering đòi hỏi sự linh hoạt
- Bảo mật phải được tích hợp, không tùy chọn
- Kiểm tra tự động nâng cao độ tin cậy
- Lập kế hoạch chi tiết là yếu tố then chốt
- Dự đoán tương lai cần thận trọng