Công cụ (Tool)

Bởi:

tonthathoaan

Công cụ (Tool)

Sử dụng một ngôn ngữ mô hình hóa phức tạp và rộng mở như UML cần thiết sự trợ giúp của công cụ. Mặc dù phác thảo đầu tiên của một mô hình có thể được thực hiện bằng bảng trắng cùng giấy và mực, nhưng công việc bảo trì, đồng bộ hóa và đảm bảo sự nhất quán trong một loạt các biểu đồ khác nhau thường lại không thể trở thành khả thi nếu không có công cụ.

Thị trường công cụ mô hình hóa đã dừng trong mức độ sơ khởi suốt một thời gian dài kể từ khi xuất hiện ý tưởng đầu tiên về các chương trình trợ giúp cho việc tạo chương trình. Rất nhiều công cụ trong thực tế chỉ thông minh hơn các chương trình vẽ một chút, sử dụng một vài quy chế kiểm tra tính nhất quán hoặc một vài kiến thức về phương pháp và ngôn ngữ mô hình hóa. Mặc dù đã có một vài bước tiến nhất định và nhiều công cụ hôm nay đã tới gần sáng kiến khởi thủy kia nhiều hơn (Rational Rose), nhưng thị trường vẫn còn không ít công cụ chưa được gọt giũa, vẫn còn chứa lỗi hoặc những nét kỳ quặc, kể cả những vấn đề đơn giản như copy và dán. Những công cụ này còn hạn chế ở phương diện rằng tất cả bọn chúng đều có ngôn ngữ mô hình hóa riêng, hay ít nhất thì cũng có những định nghĩa riêng của chúng về ngôn ngữ này.

Cùng với sự ra đời của ngôn ngữ UML, các nhà cung cấp công cụ mô hình hóa giờ đây có thể dành nhiều thời gian hơn cho việc nâng cấp công cụ, bởi họ không cần phải dồn tâm dồn sức cho việc định nghĩa các phương pháp mới cũng như các ngôn ngữ mới.

Một công cụ mô hình hóa hiện đại cần phải cung cấp các chức năng sau:

*Vẽ biểu đồ***:** cần phải tạo điều kiện dễ dàng vẽ ra các biểu đồ trong ngôn ngữ mô hình hóa. Công cụ cần phải đủ khả năng thông minh để hiểu mục đích của các biểu đồ và biết được những ngữ nghĩa cũng như các quy tắc đơn giản, đủ để nó có thể cảnh báo hoặc ngăn chặn việc sử dụng không thích hợp các phần tử mô hình.

Hoạt động như một nhà kho **(***Repository***):** công cụ cần phải hỗ trợ một nhà kho trung tâm để tất cả các thông tin về mô hình được lưu trữ trong cùng một chỗ. Nếu ví dụ tên của một lớp bị thay đổi trong một biểu đồ, thì sự thay đổi này cần phải xảy ra trong tất cả các biểu đồ khác có sử dụng lớp này.

Hỗ trợ định hướng **(***Navigation***):** công cụ cần phải tạo điều kiện dễ dàng cho người sử dụng định hướng và chuyển dịch trong mô hình để theo dõi một phần tử từ biểu đồ này sang biểu đồ khác, hoặc để mở rộng lời miêu tả của một phần tử.

*Hỗ trợ nhiều người sử dụng (multiuser support)***:** Công cụ cần hỗ trợ cho nhiều người sử dụng, và tạo điều kiện cho họ cùng làm việc với một mô hình mà không ngăn chặn hoặc quấy phá lẫn nhau.

*Tự động tạo code (code generate)***:** một công cụ cao cấp cần phải có khả năng tạo ra code, nơi tất cả các thông tin trong mô hình được chuyển tải thành các khung code (code skeletons), được sử dụng làm nền tảng cho giai đoạn xây dựng chương trình.

*Tái tạo mô hình(Reserve engineer)***:** Một công cụ cao cấp cần phải có khả năng đọc những thành phần code đang tồn tại và từ đó sản xuất ra mô hình. Từ đó suy ra, một mô hình có thể được làm từ những dòng code đã tồn tại; hoặc một nhà phát triển có thể dễ dàng chuyển đi chuyển về giữa công việc mô hình hóa và công việc lập trình.

*Tích hợp với các công cụ khác***:** một công cụ cần phải có khả năng tích hợp với những công cụ khác, với cả việc phát triển môi trường, ví dụ như các trình soạn thảo (editor), chương trình dịch (compiler), chương trình tìm lỗi (debugger) cũng như các công cụ của doanh nghiệp khác như công cụ quản trị cấu hình, hệ thống theo dõi các phiên bản.

*Bao quát mô hình ở tất cả các mức độ trừu tượng hóa khác nhau***:** công cụ cần phải dễ chuyển tải từ lời miêu tả ở cấp trừu tượng hóa cao nhất của hệ thống (tức là ở dạng một lượng các gói khác nhau) đi xuống cho tới cấp của những dòng code thật sự. Sau đó, để truy xuất những dòng lệnh code cho một thủ tục cụ thể nào đó trong một lớp nào đó, bạn có thể chỉ cần nhấp chuột vào tên của thủ tục đó trong một biểu đồ.

Trao đổi mô hình: Một mô hình hay một biểu đồ của một mô hình nào đó cần phải có khả năng được xuất ra từ một công cụ này rồi nhập vào một công cụ khác, giống như những dòng lệnh code được sản sinh trong một công cụ này có thể được sử dụng trong một công cụ khác. Nguyên tắc trao đổi đó cần phải được áp dụng cho các mô hình trong một ngôn ngữ mô hình hóa được định nghĩa chính xác.

Tóm tắt về UML

UML tổ chức một mô hình thành một loạt các hướng nhìn, thể hiện các khía cạnh khác nhau của hệ thống. Chỉ khi kết hợp tất cả các hướng nhìn lại với nhau, người ta mới co được một bức tranh trọn vẹn về hệ thống. Một hướng nhìn không phải là một hình vẽ, nội dung của nó được miêu tả qua các biểu đồ, đây là những hình vẽ chứa đựng các phần tử mô hình hóa. Một biểu đồ bình thường chỉ trình bày một phần nội dung của một hướng nhìn, và một hướng nhìn được định nghĩa với rất nhiều biểu đồ. Một biểu đồ chứa các phần tử mô hình, ví dụ như lớp, đối tượng, nút mạng, thành phần và những mối quan

hệ như nối kết, khái quát hóa, phụ thuộc. Các phần tử này có ý nghĩa (semantic) và các ký hiệu hình học.

Các loại biểu đồ trong UML là: biểu đồ lớp, biểu đồ đối tượng, biểu đồ Use case, biểu đồ trạng thái, biểu đồ trình tự, biểu đồ cộng tác, biểu đồ hành động, biểu đồ thành phần và biểu đồ triển khai. Mục đích của các loại biểu đồ cũng như quy tắc vẽ chúng sẽ được miêu tả chi tiết trong chương sau.

UML có một số cơ chế chung để bổ sung thông tin không thể được thể hiện trong quá trình vẽ biểu đồ. Những thông tin này bao gồm ví dụ những thành phần trang trí, các lời ghi chú có thể chứa bất kỳ loại thông tin nào cũng như các thuộc tính đặc tả. Ngoài ra còn có các cơ chế mở rộng, bao gồm giá trị đính kèm, hạn chế đối với phần tử, và khuôn mẫu, định nghĩa một loại phần tử mô hình mới dựa trên một phần tử sẵn có.

Một hệ thống sẽ được miêu tả trong nhiều loại mô hình khác nhau, mỗi loại mô hình nhằm một mục đích khác nhau. Mô hình phân tích miêu tả những yêu cầu về mặt chức năng và mô hình hóa các lớp ngoài đời thực. Mô hình thiết kế chuyển tải kết quả phân tích thành một giải pháp kỹ thuật, theo khái niệm của một thiết kế phần mềm hoạt động hoàn chỉnh. Mô hình xây dựng code thể hiện hệ thống qua việc thảo chương cho nó trong một ngôn ngữ lập trình hướng đối tượng. Và cuối cùng, mô hình triển khai định vị chương trình vừa được tạo nên trong một kiến trúc vật lý bao gồm các máy tính và các trang thiết bị. Công việc được làm theo nhiều vòng lặp khác nhau chứ không phải chỉ là một chuỗi thực hiện một lần.

Để sử dụng UML một cách nghiêm chỉnh cho một dự án có thật ngoài đời, bạn cần công cụ. Một công cụ tân tiến có khả năng cho người dùng vẽ biểu đồ, trữ tất cả các thông tin vào một kho chung, cho phép dễ dàng dịch chuyển giữa các hướng nhìn và biểu đồ khác nhau trong mô hình, tạo báo cáo và tài liệu, tạo khung code từ mô hình, đọc những dòng code sẵn có rồi sản sinh ra mô hình từ đó, và dễ dàng tích hợp với các công cụ phát triển khác.

PhẦn Câu hỎi

Hỏi: UML có công cụ nào giúp nắm bắt các yêu cầu của khách hàng (người sử dụng)?

Đáp: Use Case

Hỏi: Một biểu đồ trong UML có bao chứa các hướng nhìn khác nhau.

Đáp: Sai, một hướng nhìn bao gồm một loại các biểu đồ khác nhau

Hỏi: Hãy liệt kê các thành phần chủ yếu của ngôn ngữ UML

Đáp: Hướng nhìn(View), Biểu đồ (Diagram), Phần tử mô hình, Cơ chế chung.

Hỏi: UML có công cụ nào phục vụ cho giai đoạn thử nghiệm đơn vị (Unit Testing)?

Đáp: Biểu đồ lớp và đặc tả lớp

Hỏi: UML có công cụ nào phục vụ cho giai đoạn thử nghiệm hệ thống (System Testing)?

Đáp: Use case Diagram

Hỏi: UML tạo nền tảng cho việc giao tiếp giữa khách hàng, nhà phân tích, nhà thiết kế và lập trình viên.

Đáp: Đúng