



Mở rộng UML

Bởi:

duongkieuhoa

tonthathoan

Mở rộng UML

UML có thể được mở rộng hoặc có thể được sửa đổi để phù hợp với một phương pháp đặc biệt, một tổ chức cụ thể hay một người dùng cụ thể. Chúng ta sẽ bàn luận sơ qua đến ba cơ chế mở rộng UML: khuôn mẫu (stereotype), giá trị đính kèm (tagged value) và hạn chế (constraint).

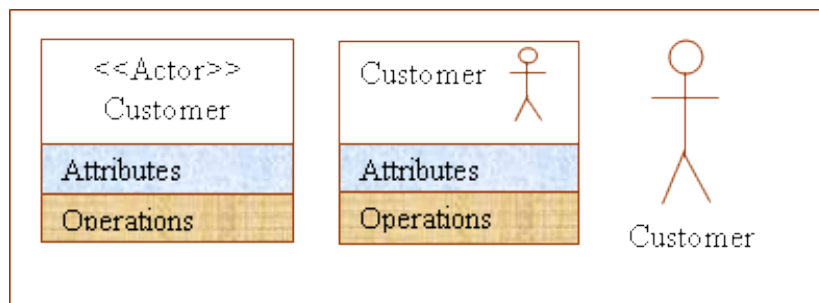
Khuôn mẫu (Stereotype)

Cơ chế mở rộng khuôn mẫu định nghĩa một loại phần tử mô hình mới dựa trên một phần tử mô hình đã tồn tại. Khuôn mẫu có thể được coi là "tương tự" như một phần tử đã có sẵn, cộng thêm phần quy định ngữ nghĩa (semantic) riêng biệt không có trong phần tử gốc kia. Khuôn mẫu của một phần tử có thể được sử dụng trong cùng tình huống như phần tử căn bản. Khuôn mẫu dựa trên tất cả các loại phần tử mô hình sẵn có - lớp, nút mạng, thành phần, cũng như các mối quan hệ như liên kết, khái quát hóa, sự phụ thuộc. Ngôn ngữ UML có chứa một số lượng lớn các khuôn mẫu được định nghĩa sẵn và chúng được sử dụng để sửa đổi các phần tử mô hình sẵn có, thay cho việc phải định nghĩa hoàn toàn mới. Cơ chế này giúp gìn giữ tính đơn giản của nền tảng ngôn ngữ UML.

Khuôn mẫu được miêu tả qua việc đưa tên của chúng vào trong một cặp ký tự ngoặc nhọn <<>>, theo như trong hình 3.16. Ký tự ngoặc nhọn này được gọi là guillements. Khuôn mẫu cũng có thể có kí hiệu hình học riêng. Một phần tử của một loại khuôn mẫu cụ thể có thể được thể hiện bởi tên khuôn mẫu đi kèm ký hiệu hình học mô tả phần tử căn bản, hay là sự kết hợp của cả hai yếu tố này. Bất kỳ khi nào một phần tử mô hình được nối kết với một tên hoặc kí hiệu khuôn mẫu, ta sẽ đọc "đây là một loại phần tử thuộc loại khuôn mẫu...". Ví dụ, một lớp với <<Window>> sẽ được gọi là "một lớp trong dạng khuôn mẫu cửa sổ", ý nghĩa của nó là một dạng lớp cửa sổ. Những thuộc tính cụ thể mà một lớp cửa sổ cần phải có sẽ được định nghĩa khi khuôn mẫu này được định nghĩa.

Như đã nói, khuôn mẫu là một cơ chế mở rộng xuất sắc, là một cơ chế ngăn cho ngôn ngữ UML không trở nên quá phức tạp, mặc dù vẫn cho phép thực hiện sự mở rộng và sửa đổi cần thiết. Đa phần các phần tử mô hình mới mà bạn cần đến đều có một khuôn

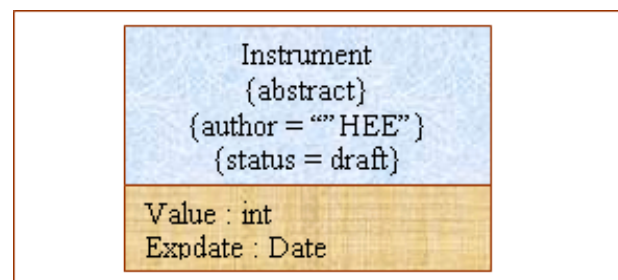
mẫu nền tảng trong ngôn ngữ UML. Một khuôn mẫu sau đó có thể được sử dụng để cộng thêm các ngữ nghĩa cần thiết, nhằm mục đích định nghĩa nên các phần tử mô hình còn thiếu.



Customer là một lớp khuôn mẫu - Actor

Giá trị đính kèm (Tagged Value)

Như đã nói, các phần tử mô hình có thể có các thuộc tính chứa một cặp tên-giá trị về bản thân chúng (hình 3.17). Các thuộc tính này cũng còn được gọi là các giá trị đính kèm. UML có chứa một loạt các thuộc tính được định nghĩa trước, nhưng kể cả người sử dụng cũng có thể định nghĩa ra các thuộc tính mới để chứa các thông tin bổ sung về các phần tử mô hình. Mọi hình dạng thông tin đều có thể được đính kèm vào phần tử: các thông tin chuyên biệt về phương pháp, các thông tin của nhà quản trị về tiến trình mô hình hóa, các thông tin được sử dụng bởi các công cụ khác, ví dụ như các công cụ tạo code, hay bất kỳ một loại thông tin nào mà người sử dụng muốn đính kèm vào phần tử mô hình.



Một ví dụ về Tagged Value

Hạn chế (Constraint)

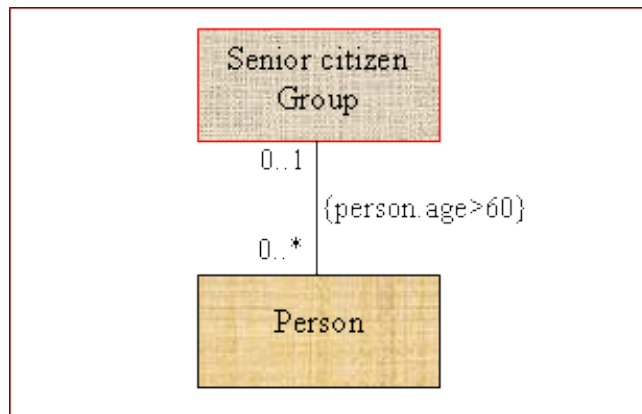
Một sự hạn chế là một sự giới hạn về sự sử dụng hoặc ý nghĩa của một phần tử. Sự hạn chế hoặc sẽ được khai báo trong công cụ và được sử dụng nhiều lần trong rất nhiều biểu đồ khác nhau, hay được định nghĩa và sử dụng trong chỉ một biểu đồ, theo như nhu cầu.

Hình 3.18 chỉ ra mối quan hệ nối kết giữa nhóm các công dân lớn tuổi và lớp con người, chỉ ra rằng nhóm công dân có thể có nhiều người liên quan. Mặc dù vậy, để miêu tả rằng chỉ những người nào lớn hơn 60 tuổi mới có thể tham gia vào nhóm này, người ta

Mở rộng UML

định nghĩa một sự hạn chế, hạn hẹp tiêu chuẩn tham gia đối với chỉ những người nào mà thuộc tính tuổi tác có giá trị lớn hơn 60. Định nghĩa này sẽ hạn chế số lượng những người được sử dụng trong mỗi quan hệ. Nếu không có nó, người ta rất dễ hiểu lầm khi diễn tả biểu đồ. Trong trường hợp tồi tệ, nó có thể dẫn đến sự thực thi sai trái của hệ thống.

Trong trường hợp này, hạn chế được định nghĩa và ứng dụng trực tiếp trong chính biểu đồ mà nó được cần tới. Nhưng nhìn chung thì hạn chế cũng có thể được định nghĩa với tên cùng lời đặc tả riêng, ví dụ như: "công dân già" và "người có tuổi lớn hơn 60", và hạn chế này sẽ được sử dụng trong nhiều biểu đồ khác nhau. UML có chứa một loạt các hạn chế được định nghĩa sẵn, chúng được miêu tả chi tiết trong các chương sau.



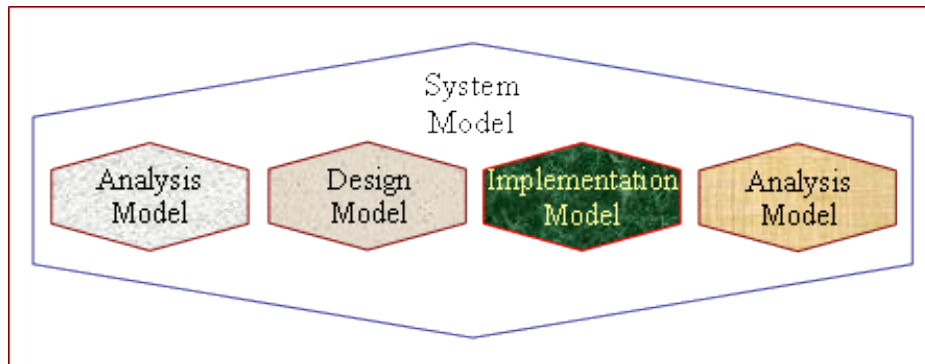
Một ràng buộc hạn chế đối tượng Person góp phần vào quan hệ kết hợp

Mô hình hóa với UML

Khi xây dựng hệ thống với UML, người ta không chỉ xây dựng duy nhất một mô hình. Sẽ có nhiều mô hình khác nhau trong những giai đoạn phát triển khác nhau, nhằm đến các mục đích khác nhau. Trong giai đoạn phân tích, mục đích của mô hình là nắm bắt tất cả các yêu cầu đối với hệ thống và mô hình hóa nền tảng bao gồm các lớp và các cộng tác "đời thực". Trong giai đoạn thiết kế, mục đích của mô hình là mở rộng mô hình phân tích, tạo thành một giải pháp kỹ thuật khả thi, có chú ý đến môi trường của công việc xây dựng (viết code). Trong giai đoạn xây dựng code, mô hình chính là những dòng code nguồn thật sự, được viết nên và được dịch thành các chương trình. Và cuối cùng, trong giai đoạn triển khai, một lời miêu tả sẽ giải thích hệ thống cần được triển khai ra sao trong kiến trúc vật lý. Khả năng theo dõi xuyên suốt nhiều giai đoạn và nhiều mô hình khác nhau được đảm bảo qua các thuộc tính hoặc các mối quan hệ nâng cao (refinement).

Mặc dù đó là các mô hình khác nhau, nhưng chúng đều được xây dựng nên để mở rộng nội dung của các mô hình ở giai đoạn trước. Chính vì thế, tất cả các mô hình đều cần phải được gìn giữ tốt để người ta có thể dễ dàng đi ngược lại, mở rộng ra hay tái thiết

lập mô hình phân tích khởi đầu và rồi dần dần từng bước đưa các sự thay đổi vào mô hình thiết kế cũng như các mô hình xây dựng (hình 3.19).

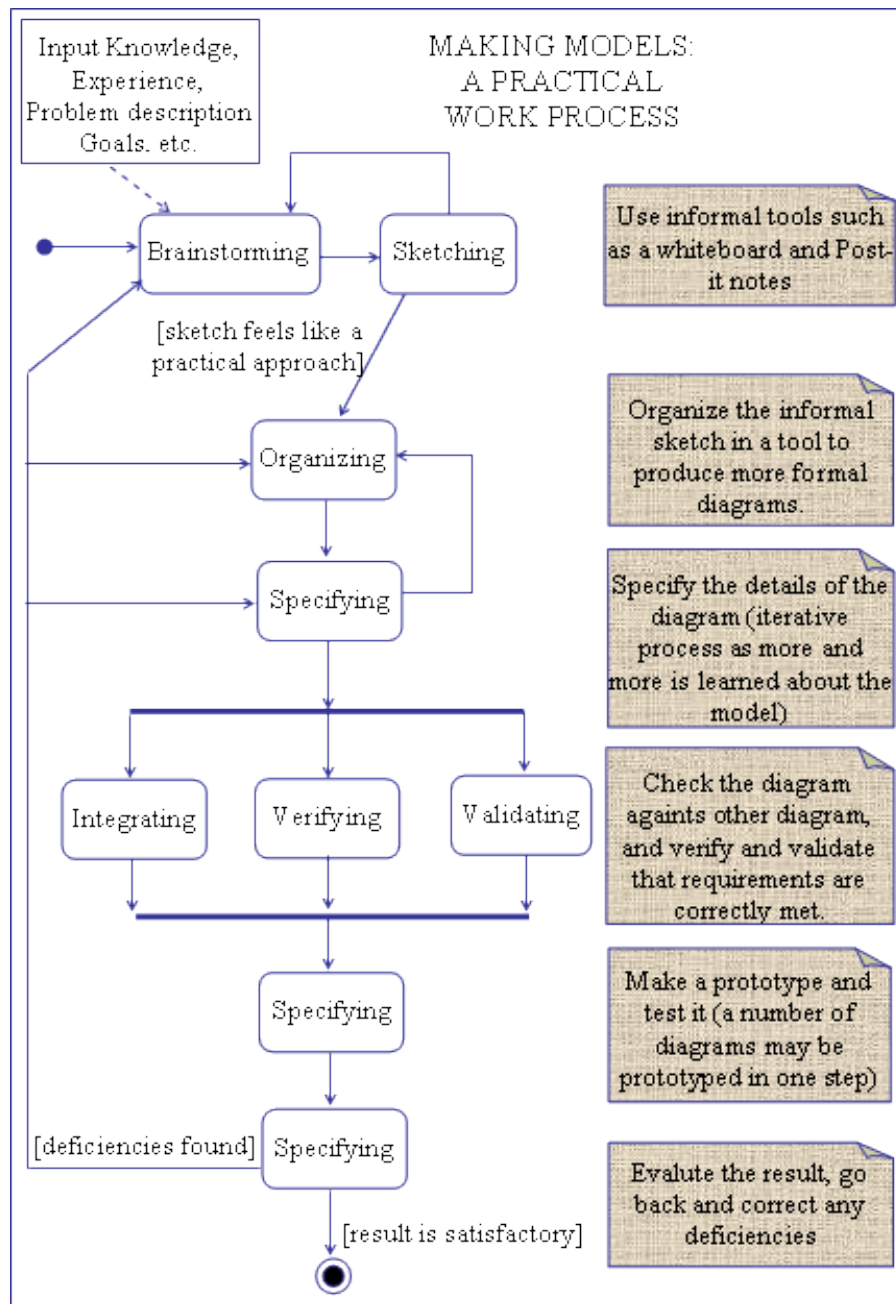


Một hệ thống được mô tả trong nhiều mô hình

Bản thân ngôn ngữ UML không phụ thuộc vào giai đoạn, có nghĩa là cũng những nguyên tắc ngôn ngữ đó và cũng những biểu đồ đó được sử dụng để mô hình hóa những sự việc khác nhau trong những giai đoạn khác nhau. Nhà thiết kế nắm quyền quyết định xem một mô hình sẽ phải thay đổi nhằm đạt được những mục đích nào và bao trùm những phạm vi nào. Ngôn ngữ mô hình hóa chỉ cung cấp khả năng để tạo ra các mô hình trong một phong cách mở rộng và nhất quán.

Khi mô hình hóa bằng ngôn ngữ UML, toàn bộ công việc cần phải được thực hiện theo một phương pháp hay một qui trình, xác định rõ những bước công việc nào phải được tiến hành và chúng phải được thực thi ra sao. Một qui trình như vậy thường sẽ chia công việc ra thành các vòng lặp kế tiếp, mỗi vòng lặp bao gồm các công việc: *phân tích yêu cầu/ phân tích/ thiết kế/ thực hiện/ triển khai*. Mặc dù vậy, cũng có một quy trình nhỏ hơn đề cập tới nội dung của việc mô hình hóa. Bình thường ra, khi sản xuất một mô hình hoặc sản xuất chỉ một biểu đồ duy nhất, công việc sẽ bắt đầu bằng việc thu thập một nhóm thích hợp các cá nhân khác nhau, trình bày vấn đề và mục tiêu; họ cộng tác cho một giai đoạn hội thảo khoa học và phác thảo, trao đổi những sáng kiến và ý tưởng về mô hình có thể. Công cụ được sử dụng trong giai đoạn này là hết sức khác biệt và mang tính ngẫu hứng - thường là giấy dán *post it* hay bảng trắng. Công việc được quyết định chừng nào những người tham gia có cảm giác họ đã có được một nền tảng thực tiễn cho một mô hình (giống như một tiêu đề). Kết quả sau đó sẽ được đưa vào một công cụ, mô hình tiêu đề được tổ chức, và sau đó một biểu đồ thực sự sẽ được tạo dựng nên, phù hợp với những quy định của ngôn ngữ mô hình hóa. Sau đó, mô hình được chi tiết hóa qua những công việc mang tính vòng lặp, càng ngày càng có nhiều chi tiết về giải pháp được phát hiện, được dữ liệu hóa và được bổ sung. Khi đã có nhiều thông tin hơn được thu thập về vấn đề cũng như giải pháp của nó, tiêu đề ban đầu dần dần trở thành một lời chuẩn đoán cho một mô hình có khả năng sử dụng. Khi mô hình đã gần hoàn thiện, một sự tích hợp và thẩm định sẽ được thực hiện, dẫn tới việc mô hình hoặc biểu đồ sẽ được tích hợp với những mô hình và biểu đồ khác trong cùng dự án để đảm bảo sự nhất quán.

Mô hình sau đó cũng được kiểm tra lại để chắc chắn nó đang giải quyết đúng vấn đề cần giải quyết (hình 3.20).



Một tiến trình cho công việc mô hình hoá thực tế

Cuối cùng, mô hình sẽ được thực thi và triển khai thành một loạt các nguyên mẫu (prototype), nguyên mẫu này sẽ được kiểm tra để tìm khiếm khuyết. Các khiếm khuyết bao gồm kể cả các chức năng còn thiếu, sự thực hiện tồi tệ hay phí sản xuất và phát triển quá cao. Những khiếm khuyết thường sẽ ép nhà phát triển rà đi rà lại công việc của mình để khắc phục chúng. Nếu vấn đề là quá lớn, nhà phát triển có thể sẽ đi ngược lại tất cả các bước công việc của mình cho tới tận giai đoạn sơ phác đầu tiên. Nếu các vấn đề này không lớn, nhà phát triển có lẽ chỉ cần thay đổi một vài thành phần trong tổ chức hoặc

đặc tả của mô hình. Xin nhớ rằng bước tạo nguyên mẫu không thể được thực hiện ngay lập tức sau khi hoàn tất biểu đồ; nó chỉ nên được thực hiện khi đã có một số lượng lớn các biểu đồ liên quan. Nguyên mẫu sau này có thể được vứt đi, có thể được tạo dựng nên chỉ để nhằm mục đích kiểm tra, hoặc là nếu bước tạo nguyên mẫu này thành công, nó sẽ trở thành một vòng lặp trong quy trình phát triển thật sự.