



Thiết kế chi tiết các biểu đồ lớp

Bởi:

Đoàn Văn Ban

Mục đích của UML

Mục đích chính của UML:

1. Mô hình được các hệ thống (không chỉ hệ thống phần mềm) và sử dụng được tất cả các khái niệm hướng đối tượng một cách thống nhất.
2. Cho phép đặc tả, hỗ trợ để đặc tả tường minh (trực quan) mối quan hệ giữa các khái niệm cơ bản trong hệ thống, đồng thời mô tả được mọi trạng thái hoạt động của hệ thống đối tượng. Nghĩa là cho phép mô tả được cả mô hình tĩnh lẫn mô hình động một cách đầy đủ và trực quan.
3. Tận dụng được những khả năng sử dụng lại và kế thừa ở phạm vi diện rộng để xây dựng được những hệ thống phức tạp và nhạy cảm như: các hệ thống động, hệ thống thời gian thực, hệ thống nhúng thời gian thực, v.v.
4. Tạo ra những ngôn ngữ mô hình hoá sử dụng được cho cả người lẫn máy tính.

Tóm lại, UML là ngôn ngữ mô hình hoá, ngôn ngữ đặc tả và ngôn ngữ xây dựng mô hình trong quá trình phát triển phần mềm, đặc biệt là trong phân tích và thiết kế hệ thống hướng đối tượng. UML là ngôn ngữ hình thức, thống nhất và chuẩn hoá mô hình hệ thống một cách trực quan. Nghĩa là các thành phần trong mô hình được thể hiện bởi các ký hiệu đồ hoạ, biểu đồ và thể hiện đầy đủ mối quan hệ giữa các chúng một cách thống nhất và có logic chặt chẽ.

Tuy nhiên cũng cần lưu ý:

- UML không phải là ngôn ngữ lập trình, nghĩa là ta không thể dùng UML để viết chương trình. Nó cũng không phải là một công cụ CASE. Một số công cụ CASE như Rational Rose [17] sử dụng mô hình UML để phát sinh mã nguồn tự

động sang những ngôn ngữ lập trình được lựa chọn như C++, Java, Visual C++, v.v.

- UML cũng không phải là một phương pháp hay một quá trình phát triển phần mềm. Các ký hiệu UML được sử dụng trong các dự án phát triển phần mềm nhằm áp dụng những cách tiếp cận khác nhau cho quá trình phát triển phần mềm nhằm tách chu kỳ phát triển hệ thống thành những hoạt động, các tác vụ, các giai đoạn và các bước khác nhau.

Quá trình phát triển phần mềm thống nhất với UML

- UML được phát triển để đặc tả quá trình phát triển phần mềm, nhằm mô hình hoá hệ thống. Quá trình phát triển phần mềm này gọi là *quá trình phát triển phần mềm hợp nhất* (USPD) hay quá trình hợp nhất Rational (RUP [17, 21]), gọi tắt là *quá trình hợp nhất* (UP).

RUP là tập các qui tắc hướng dẫn về phương diện kỹ thuật và tổ chức để phát triển phần mềm, nhấn mạnh chủ yếu vào các bước phân tích và thiết kế.

RUP được cấu trúc theo hai chiều:

1. *Chiều thời gian*: chia quá trình thành các pha thực hiện và các bước lặp.
2. *Chiều thành phần*: các sản phẩm cùng với các hoạt động được xác định đầy đủ.

1. Cấu trúc dự án theo chiều thời gian bao gồm các pha thực hiện:

(i) *Khởi động (Inception)*: xác định dự án tổng thể

(ii) *Soạn thảo dự án tỉ mỉ (Elaboration)*:

- + Lập kế hoạch cho những hoạt động cần thiết
- + Xác định những tài nguyên cần để thực hiện dự án
- + Xác định các tính chất, đặc trưng của dự án
- + Xây dựng kiến trúc cho hệ thống.

(iii) *Xác định những sản phẩm ở mỗi pha thực hiện.*

(iv) *Chuyển giao*: cung cấp sản phẩm cho cộng đồng người sử dụng.

2. Cấu trúc dự án theo chiều thành phần bao gồm các hoạt động:

- *Mô hình hoá nghiệp vụ*: thiết lập các khả năng của hệ thống cần xây dựng và nhu cầu của NSD.
- *Phân tích các yêu cầu*: chi tiết các yêu cầu chức năng và phi chức năng của hệ thống.
- *Phân tích thiết kế hệ thống*: mô tả hệ thống thực hiện các yêu cầu và hỗ trợ cài đặt.
- *Cài đặt chương trình*: lập trình những kết quả thiết kế nêu trên để hệ thống hoạt động đúng theo yêu cầu.
- *Kiểm thử, kiểm chứng các thành phần và toàn bộ hệ thống*.
- *Triển khai hệ thống*: khai thác hệ thống và huấn luyện NSD.

UP bao gồm con người, dự án, sản phẩm, qui trình và công cụ. Con người là những người tham gia dự án để tạo ra sản phẩm phần mềm theo một quá trình với sự hỗ trợ của công cụ được cung cấp.

UP là quá trình phát triển phần mềm được *hướng dẫn bởi các ca sử dụng*. Nghĩa là các yêu cầu của NSD được mô tả trong các ca sử dụng, là chuỗi các hành động được thực hiện bởi hệ thống nhằm cung cấp các dịch vụ, các thông tin cho khách hàng. Các ca sử dụng bao gồm chuỗi các công việc được xem là nền tảng để tạo ra mô hình thiết kế và cài đặt hệ thống.

UP cũng là qui trình tập trung vào kiến trúc, được lập và *phát triển tăng trưởng liên tục*. Kiến trúc của hệ thống phải được thiết kế nhằm đáp ứng các yêu cầu của các ca sử dụng chính, trong giới hạn của chuẩn phần cứng mà hệ thống sẽ chạy và của cấu trúc của cả hệ thống lẫn các hệ thống con. Tính lặp của quá trình phát triển phần mềm được thể hiện ở chỗ là một dự án được chia thành các dự án nhỏ và được thực hiện lặp lại trong từng bước thực hiện. Mỗi dự án nhỏ đều thực hiện phân tích, thiết kế, cài đặt và kiểm thử, v.v. Mỗi phần việc đó được phát triển tăng trưởng và cả dự án cũng được thực hiện theo sự tăng trưởng này.

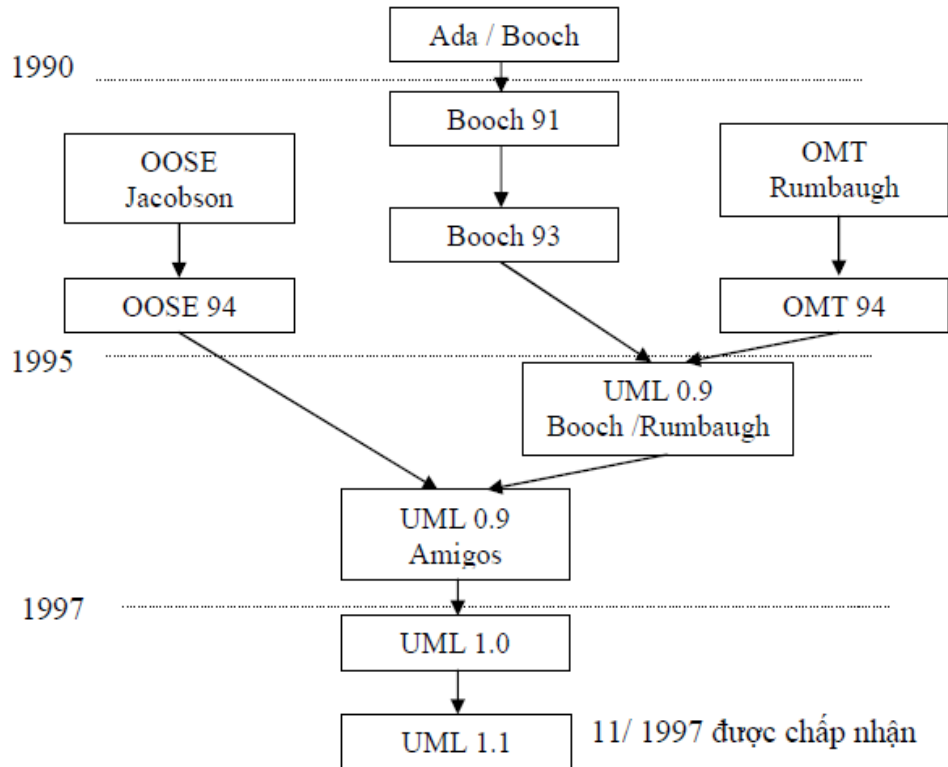
UP không chỉ tạo ra một hệ thống phần mềm hoàn chỉnh mà còn tạo ra một số sản phẩm trung gian như các mô hình. Các mô hình chính trong UP là mô hình nghiệp vụ (ca sử dụng), mô hình khái niệm, mô hình thiết kế, mô hình triển khai và mô hình thử nghiệm. Các mô hình này có sự phụ thuộc theo vết phát triển, nghĩa là có thể lần theo từng mô hình để đến được mô hình trước.

Giới thiệu tổng quát về UML

UML được xây dựng dựa chính vào:

- *Cách tiếp cận của Booch* (Booch Approach),
- *Kỹ thuật mô hình đối tượng* (OMT – Object Modeling Technique) của Rumbaugh,

- Công nghệ phần mềm hướng đối tượng (OOSE – Object-Oriented Software Engineering) của Jacobson,
- Đồng thời thống nhất được nhiều kỹ pháp, khái niệm của các phương pháp khác. Quá trình hình thành UML bắt đầu từ ngôn ngữ Ada (Booch) trước năm 1990 (hình 2-1).

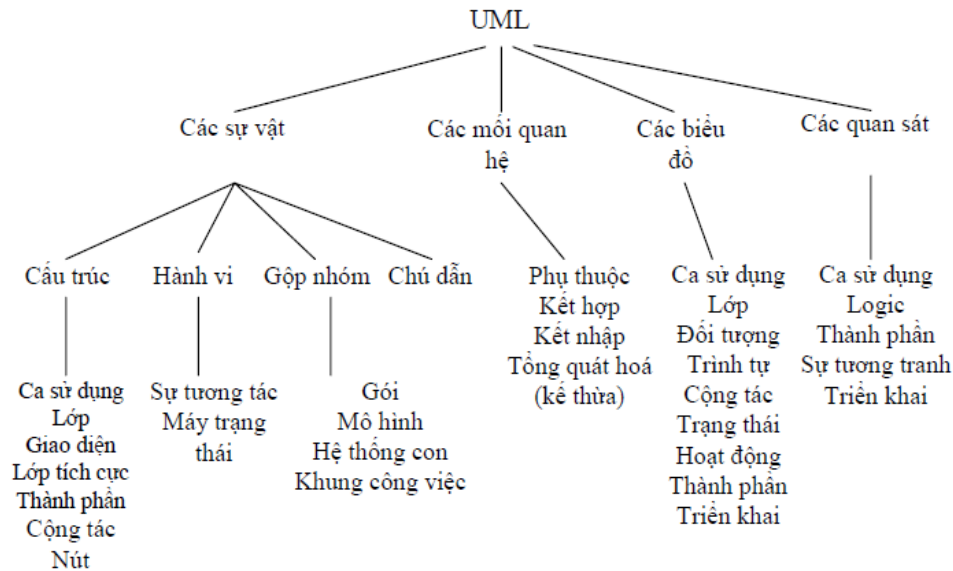


Sự phát triển của UML

Để hiểu và sử dụng tốt UML trong phân tích, thiết kế hệ thống, đòi hỏi phải nắm bắt được ba vấn đề chính:

1. Các phần tử cơ bản của UML,
2. Những quy định liên kết giữa các phần tử, các quy tắc cú pháp,
3. Những cơ chế chung áp dụng cho ngôn ngữ mô hình hoá hệ thống.

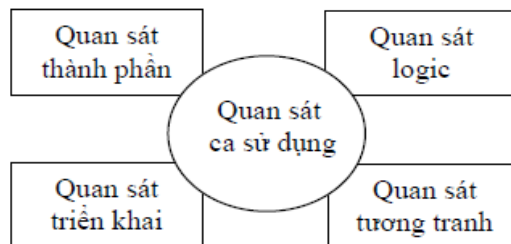
Các phần tử của UML



Các thành phần cơ sở của UML

Các quan sát

Các quan sát (góc nhìn) theo các phương diện khác nhau của hệ thống cần phân tích, thiết kế. Dựa vào các quan sát để thiết lập kiến trúc cho hệ thống cần phát triển. Có năm loại quan sát: *quan sát theo ca sử dụng*, *quan sát logic*, *quan sát thành phần*, *quan sát tương tranh* và *quan sát triển khai*. Mỗi quan sát tập trung khảo sát và mô tả một khía cạnh của hệ thống (hình 3) và thường được thể hiện trong một số biểu đồ nhất định.



Các quan sát của hệ thống

- *Quan sát các ca sử dụng (hay trường hợp sử dụng)*: mô tả các chức năng, nhiệm vụ của hệ thống. Quan sát này thể hiện mọi yêu cầu của hệ thống, do vậy nó phải được xác định ngay từ đầu và nó được sử dụng để điều khiển, thúc đẩy và thẩm định hay kiểm tra các công việc của tất cả các giai đoạn của cả quá trình phát triển phần mềm. Nó cũng là cơ sở để trao đổi giữa các thành viên của dự án phần mềm và với khách hàng. Quan sát ca sử dụng được thể hiện trong các *biểu đồ ca sử* và có thể ở một vài *biểu đồ trình tự*, *cộng tác*, v.v.
- Quan sát logic biểu diễn cách tổ chức logic của các lớp và các quan hệ của chúng với nhau. Nó mô tả cấu trúc tĩnh của các lớp, đối tượng và sự liên hệ của chúng thể hiện mối liên kết động thông qua sự trao đổi các thông điệp. Quan

sát được thể hiện trong các biểu đồ lớp, biểu đồ đối tượng, biểu đồ tương tác, biểu đồ biến đổi trạng thái. Quan sát logic tập trung vào cấu trúc của hệ thống. Trong quan sát này ta nhận ra các bộ phận cơ bản cấu thành hệ thống thể hiện mọi quá trình trao đổi, xử lý thông tin cơ bản trong hệ thống.

- Quan sát thành phần (quan sát cài đặt) xác định các mô đun vật lý hay tệp mã chương trình và sự liên hệ giữa chúng để tổ chức thành hệ thống phần mềm. Trong quan sát này ta cần bổ sung: chiến lược cấp phát tài nguyên cho từng thành phần, và thông tin quản lý như báo cáo tiến độ thực hiện công việc, v.v. Quan sát thành phần được thể hiện trong các biểu đồ thành phần và các gói.
- Quan sát tương tranh (quan sát tiến trình) biểu diễn sự phân chia các luồng thực hiện công việc, các lớp đối tượng cho các tiến trình và sự đồng bộ giữa các luồng trong hệ thống. Quan sát này tập trung vào các nhiệm vụ tương tranh, tương tác với nhau trong hệ thống đa nhiệm.
- Quan sát triển khai mô tả sự phân bổ tài nguyên và nhiệm vụ trong hệ thống. Nó liên quan đến các tầng kiến trúc của phần mềm, thường là kiến trúc ba tầng: tầng giao diện (tầng trình diễn hay tầng sử dụng), tầng logic tác nghiệp và tầng lưu trữ CSDL được tổ chức trên một hay nhiều máy tính khác nhau. Quan sát triển khai bao gồm các luồng công việc, bộ xử lý và các thiết bị. Biểu đồ triển khai mô tả các tiến trình và chỉ ra những tiến trình nào trên máy nào.

Các sự vật (các phần tử của mô hình)

UML có bốn phần tử mô hình, đó là cấu trúc, hành vi, nhóm và chú thích.

- *Phần tử cấu trúc*: là các danh từ trong mô hình UML, biểu diễn cho các thành phần khái niệm hay vật lý của hệ thống. UML có bảy phần tử cấu trúc được mô tả như sau:

+ **Lớp**. Lớp là tập các đối tượng cùng chia sẻ với nhau về các thuộc tính, thao tác, quan hệ và ngữ nghĩa.

+ **Giao diện**. Giao diện là tập các thao tác làm dịch vụ cho lớp hay thành phần. Giao diện mô tả hành vi quan sát được từ bên ngoài thành phần. Giao diện chỉ khai báo các phương thức xử lý nhưng không định nghĩa nội dung thực hiện. Nó thường không đứng một mình mà thường được gắn với lớp hay một thành phần.

+ **Phần tử cộng tác**. Phần tử cộng tác mô tả ngữ cảnh của sự tương tác trong hệ thống. Nó thể hiện một giải pháp thi hành trong hệ thống, bao gồm các lớp, quan hệ và sự tương tác giữa chúng để thực hiện một ca sử dụng như mong đợi.

+ **Ca sử dụng**. Ca sử dụng mô tả một tập các hành động mà hệ thống sẽ thực hiện để phục vụ cho các tác nhân ngoài. Tác nhân ngoài là những gì bên ngoài có tương tác, trao đổi với hệ thống.

Thiết kế chi tiết các biểu đồ lớp

+ **Lớp tích cực.** Lớp tích cực được xem như là lớp có đối tượng làm chủ một hay nhiều tiến trình, luồng hành động.

+ **Thành phần.** Thành phần biểu diễn vật lý mã nguồn, các tệp nhị phân trong quá trình phát triển hệ thống.

+ **Nút.** Nút thể hiện thành phần vật lý tồn tại khi chương trình chạy và biểu diễn cho các tài nguyên được sử dụng trong hệ thống.

- *Phần tử mô tả hành vi:* là các động từ của mô hình, biểu diễn hành vi trong sự tương tác của các thành phần và sự biến đổi trạng thái của hệ thống. Có hai loại chính là sự tương tác và máy biến đổi trạng thái.

+ **Sự tương tác.** *Sự tương tác* là hành vi bao gồm một tập các thông điệp trao đổi giữa các đối tượng trong một ngữ cảnh cụ thể để thực hiện một ca sử dụng.

+ **Máy biến đổi trạng thái.** *Máy biến đổi trạng thái (ôôtômát hữu hạn trạng thái)* chỉ ra trật tự thay đổi trạng thái khi các đối tượng hay sự tương tác sẽ phải đi qua để đáp ứng các sự kiện xảy ra.

- *Phần tử nhóm:* là bộ phận tổ chức của mô hình UML. Phần tử nhóm có gói, mô hình và khung công việc.

+ **Gói (package).** Gói là phần tử đa năng được sử dụng để tổ chức các lớp, hay một số nhóm khác vào trong một nhóm. Không giống với thành phần (component), phần tử gói hoàn toàn là khái niệm, có nghĩa là chúng chỉ tồn tại trong mô hình vào thời điểm phát triển hệ thống chứ không tồn tại vào thời điểm chạy chương trình. Gói giúp ta quan sát hệ thống ở mức tổng quát.

+ **Mô hình.** Mô hình là những mô tả về các đặc tính tĩnh và/hoặc động của các chủ thể trong hệ thống.

+ **Khung công việc.** *Khung công việc* là một tập các lớp trừu tượng hay cụ thể được sử dụng như là các khuôn mẫu để giải quyết một họ các vấn đề tương tự.

Là bộ phận chú giải của mô hình, giải thích về các phần tử, khái niệm và cách sử dụng chúng trong mô hình.

Các mối quan hệ

UML cho phép biểu diễn cả bốn mối quan hệ giữa các đối tượng trong các hệ thống. Đó là các quan hệ: phụ thuộc, kết hợp, tổng quát hoá và hiện thực hoá.

+ **Quan hệ phụ thuộc.** Đây là quan hệ ngữ nghĩa giữa hai phần tử, trong đó sự thay đổi của một tử sẽ tác động đến ngữ nghĩa của phần tử phụ thuộc.

+ **Quan hệ kết hợp.** Kết hợp là quan hệ cấu trúc xác định mối liên kết giữa các lớp đối tượng. Khi có một đối tượng của lớp này gửi/nhận thông điệp đến/từ chỗ đối tượng của lớp kia thì hai lớp đó có quan hệ kết hợp. Một dạng đặc biệt của quan hệ kết hợp là *quan hệ kết nhập*, biểu diễn mối quan hệ giữa toàn thể và bộ phận.

+ **Quan hệ tổng quát hoá.** Đây là quan hệ mô tả sự khái quát hoá mà trong đó một số đối tượng cụ thể (của lớp con) sẽ được kế thừa các thuộc tính, các phương thức của các đối tượng tổng quát (lớp cơ sở).

+ **Hiện thực hoá.** Hiện thực hoá là quan hệ ngữ nghĩa giữa giao diện và lớp (hay thành phần) để thực hiện cài đặt các dịch vụ đã được khai báo trong các giao diện.

Các biểu đồ

Biểu đồ là đồ thị biểu diễn đồ họa về tập các phần tử trong mô hình và mối quan hệ của chúng. Biểu đồ chứa đựng các nội dung của các quan sát dưới các góc độ khác nhau, một thành phần của hệ thống có thể xuất hiện trong một hay nhiều biểu đồ. UML cung cấp những biểu đồ trực quan để biểu diễn các khía cạnh khác nhau của hệ thống, bao gồm:

- *Biểu đồ ca* sử dụng mô tả sự tương tác giữa các tác nhân ngoài và hệ thống thông qua các ca sử dụng. Các ca sử dụng là những nhiệm vụ chính, các dịch vụ, những trường hợp sử dụng cụ thể mà hệ thống cung cấp cho người sử dụng và ngược lại.
- *Biểu đồ lớp* mô tả cấu trúc tĩnh, mô tả mô hình khái niệm bao gồm các lớp đối tượng và các mối quan hệ của chúng trong hệ thống hướng đối tượng.
- *Biểu đồ trình tự* thể hiện sự tương tác của các đối tượng với nhau, chủ yếu là trình tự gửi và nhận thông điệp để thực thi các yêu cầu, các công việc theo thời gian.
- *Biểu đồ cộng tác* tương tự như biểu đồ trình tự nhưng nhấn mạnh vào sự tương tác của các đối tượng trên cơ sở cộng tác với nhau bằng cách trao đổi các thông điệp để thực hiện các yêu cầu theo ngữ cảnh công việc.
- *Biểu đồ trạng thái* thể hiện chu kỳ hoạt động của các đối tượng, của các hệ thống con và của cả hệ thống. Nó là một loại ô tômat hữu hạn trạng thái, mô tả các trạng thái, các hành động mà đối tượng có thể có và các sự kiện gắn với các trạng thái theo thời gian.
- *Biểu đồ hành động* chỉ ra dòng hoạt động của hệ thống, bao gồm các trạng thái hoạt động, trong đó từ một trạng thái hoạt động sẽ chuyển sang trạng thái khác sau khi một hoạt động tương ứng được thực hiện. Nó chỉ ra trình tự các bước,

Thiết kế chi tiết các biểu đồ lớp

tiến trình thực hiện cũng như các điểm quyết định và sự rẽ nhánh theo luồng sự kiện.

- Biểu đồ thành phần chỉ ra cấu trúc vật lý của các thành phần trong hệ thống, bao gồm: các thành phần mã nguồn, mã nhị phân, thư viện và các thành phần thực thi.
- Biểu đồ triển khai chỉ ra cách bố trí vật lý các thành phần theo kiến trúc được thiết kế của hệ thống.

Các khái niệm cơ bản của biểu đồ và cách xây dựng các biểu đồ trên để phân tích, thiết kế hệ thống sẽ được giới thiệu chi tiết ở các chương sau.