

Trong các lược đồ quen thuộc thì các cái tên như Deployment Diagram, Architecture Diagram sẽ là những lược đồ mà sẽ ít ai biết tới hoặc có chăng cũng chỉ là nghe qua! Thực tế thì chúng là những lược đồ quan trọng chẳng kém gì đàn em kia (Nói đàn em vì những loại lược đồ này mang tính khái quát cao cung như rất quan trọng trong dự án thực tế)

Trong bài viết này chúng ta sẽ cùng tìm hiểu về ông kỵ Deployment Diagram, suy cho cùng thì giá trị của một dự án chỉ thực sự được định mức khi đã deploy và có người dùng mà :v

I. Ôn lại một chút Trong UML, bạn sử dụng Class Diagram và Component Diagram để lý giải về cấu trúc của phần mềm; Sequence Diagram, Collaboration Diagram, Statechart Diagram và Activity Diagram để xác định hành vi của phần mềm.

Bên cạnh đó, phần giáp ranh giữa phần cứng và phần mềm của hệ thống thì ta dùng Deployment Diagram để giải thích sự liên kết giữa bộ xử lý <Ở đây có thể là CPU thuê từ máy ảo,...> và các thiết bị trong hệ thống mà phần mềm của chúng ta sẽ chạy ở trên đó.,>

II. Deployment Diagram là gì? Được sử dụng để biểu đạt trực quan những khía cạnh tĩnh của các nút vật lý và các mối quan hệ giữa chúng cũng như chỉ ra các chi tiết trong việc xây dựng. Nó là một trong hai loại lược đồ sử dụng trong việc mô hình hóa các khía cạnh vật lý của một hệ thống hướng đối tượng. Một lược đồ triển khai là một lược đồ cho thấy cấu hình runtime của các node đang xử lý và các components bên trong chúng. Hmmm nghe qua thì nó khác là giống với lược đồ component ở điểm node là tập hợp của nhiều component.

Class vs Node vs Component Component Diagram và Deployment Diagram có cách vẽ tựa như Class Diagram, chỉ có điều thay vì chứa các Classes, tại nó sẽ chứa các Component và các Node theo trình tự riêng. Về cơ bản thì Deployment Diagram có thể xem như Class Diagram nhưng ở đây nó tập trung và các node của hệ thống. Ta sử dụng Deployment Diagram để mô hình hóa view hệ thống khi triển khai tĩnh

Thường thì chúng ta sử dụng các lớp để mô hình hóa các khái niệm và các thứ khác trong một *problem domain* (giải thích nguyên văn: <https://bit.ly/2WasrwZ>) và sau đó lại mô hình hóa chúng với concrete instance (concrete ở đây có thể hiểu là real) bằng cách sử dụng components và nodes. **Ví dụ:** Nếu muốn mô hình hóa cấu trúc liên kết network tổ chức của bạn, bạn sẽ sử dụng Deployment Diagram chứa các Instance của các Node. Tương tự nếu bạn muốn mô hình hóa các Component sống trong các nút vật lý trong mạng này, bạn sẽ sử dụng Component Diagram chứa các Instance của các Component.

III Dăm ba khái niệm “dễ nhớ”
a. Component Một Component là tập hợp của nhiều Class có liên quan với nhau. Một số Component chỉ tồn tại tại thời điểm compile time, số khác tồn tại tại link time, và cũng có một số là ở thời điểm run-time; Tuy nhiên vẫn có trường hợp xuất hiện nhiều hơn một – Những đặc điểm trên sẽ giúp ta phân loại các Component với nhau.

b. Node Nút là một đối tượng vật lý ở thời điểm run-time, đại diện cho một tài nguyên điện toán (có bộ nhớ và khả năng xử lý). Với Deployment Diagram, ta có thể đặc tả các loại Node và các Node Instance cũng như có thể mô hình hóa các Component Instance đang chạy / “sống” trên một Node bằng cách vẽ chúng trong Node. Ngoài ra để diễn đạt communication giữa các Node ta sẽ dùng bằng đường thẳng quan hệ “Connection”.

c. Dependency Mối quan hệ ràng buộc chỉ ra rằng một *model element* (**Source**) phụ thuộc vào *another model element* (**Target**) sao cho một thay đổi đối với “*Target Element*” kéo theo sự thay đổi ở “*Source Element*” trong Dependency. Trong Deployment Diagram, ta đặc tả ràng buộc nhằm chỉ ra khả năng mà một Node có thể hỗ trợ cho một Component và tất nhiên là cho thấy ràng buộc giữa chúng.

d. Connection Nhằm đặc tả đường thông tin sử dụng bởi các phần cứng, thông thường thì đường này chỉ ra các phương thức giao tiếp ví dụ như TCP/IP.

e. Artifact Đại diện cho các Concrete Element. Vd: File thực thi, thư viện, file nén, DB Schema, file cấu hình, etc,...

IV. Nơi nào đó? Có em đây? – Deployment Diagram Khi bạn muốn xem một mô hình tổng quan của việc triển khai (ở trạng thái tĩnh) của một hệ thống, thông thường bạn sẽ sử dụng sơ đồ triển khai theo một trong ba loại hệ thống:

V. Các bước để vẽ Deployment Diagram
B1: Xác định các node sẽ hiện diện ở client và server processor của hệ thống và highlight những thiết bị có liên quan tới hành vi của hệ thống. *Ví dụ: Bạn muốn đặc tả các thiết bị đặc biệt, chẳng hạn máy đọc thẻ tín dụng, máy đọc tem giá, các hiển thị trên các thiết bị khác ngoài màn hình, bởi vì vị trí của chúng trong cấu trúc liên kết phần cứng của hệ thống có ý nghĩa về mặt kiến trúc.*

B2: Cung cấp tín hiệu trực quan cho các bộ xử lý (processor) và thiết bị này thông qua việc rập khuôn (stereotyping).

B3: Mô hình cấu trúc liên kết của các nút này trong sơ đồ triển khai

B4: Tương tự, chỉ định mối quan hệ giữa các Components trong chế độ xem implementation của hệ thống của bạn và các Nodes trong chế độ xem deployment của hệ thống.

VI. Kế hoạch triển khai Một mô hình triển khai (Deployment Model) có thể được phát triển theo các bước dưới đây:
B1: Thiết kế và lên kế hoạch cách hệ thống được cài đặt
B2: Xác định xem nếu các phiên bản khác nhau của hệ thống sẽ được triển khai cùng lúc, bằng cách nào ta có thể xử lý chúng?
B3: Physical sites nào cần triển khai và theo thứ tự nào?
B4: Cách các bạn train user?
B5: Các backups nào cần thực hiện trước khi cài đặt?

VII. Loạt các ví dụ Một Component có thể hiểu như một Code Module. Các Component Diagram là các physical analog của Class Diagram. Deployment Diagrams cho thấy các Physical Configuration của Software và Hardware.

VD1: Sơ đồ triển khai sau đây cho thấy mối quan hệ giữa các Component phần mềm và phần cứng liên quan đến các giao dịch bất động sản.

Xem thêm: tai game sam

Các Physical Hardware được tạo thành từ các Node. Mỗi Component thuộc về một Node. Các Component được hiển thị dưới dạng hình chữ nhật với hai tab ở phía trên bên trái

VD2: Hệ thống thông tin quản lý dành cho SME (doanh nghiệp vừa và nhỏ) sử dụng TCP / IP làm giao thức truyền thông mạng để kết nối giữa cơ sở dữ liệu máy chủ web, ứng dụng, máy chủ email, v.v. được bảo vệ bởi tường lửa giữa client và server side của hệ thống.

Chuyên mục: Tổng hợp

The post [Các Biểu Đồ Triển Khai Trong Uml, Bản Vẽ Triển Khai](https://webdoithuong.com/cac-bieu-do-trien-khai-trong-uml-ban-ve-trien-khai/) appeared first on [WEBDOITHUONG](https://webdoithuong.com).

Nguồn: WEBDOITHUONG

<https://webdoithuong.com/cac-bieu-do-trien-khai-trong-uml-ban-ve-trien-khai/>