**ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH**

**Khoa Công Nghệ Thông Tin**



**BÀI BÁO CÁO**

**TÌM HIỂU VỀ STATE DIAGRAM**

**MÔN PHÂN TÍCH VÀ THIẾT KẾ HƯỚNG ĐỐI TƯỢNG**

**Giảng viên hướng dẫn: Lương Trần Ngọc Khiết**

**Nhóm sinh viên thực hiện:**

**1. 4501104061 – Nguyễn Văn Giàu**

**2. 4501104134 – Huỳnh Bảo Lộc**

**3. 4501104184 – Trần Thế Phương**

**4. 4501104239 – Đỗ Tín**

**4. 4501104273 – Nguyễn Minh Văn**

**Thành phố Hồ Chí Minh, Tháng 10 / 2021**

Mục lục

[**I.** **GIỚI THIỆU VỀ STATE DIAGRAM** 3](#_Toc86259096)

[**1.** **State Diagram là gì?** 3](#_Toc86259097)

[**2.** **State Diagram dùng để làm gì?** 3](#_Toc86259098)

[**3.** **Khi nào thì sử dụng State Diagram trong UML** 3](#_Toc86259099)

[**4.** **Vị trí của State Diagram trong mô hình thiết kế hệ** 4](#_Toc86259100)

[**II.** **CÁC ĐỐI TƯỢNG VẼ TRONG POWER DESIGNER** 4](#_Toc86259101)

[**III.** **Ý NGHĨA, ĐIỀU KIỆN THIẾT KẾ STATE DIAGRAM** 7](#_Toc86259102)

[**IV.** **TÀI LIỆU THAM KHẢO** 10](#_Toc86259103)

# **GIỚI THIỆU VỀ STATE DIAGRAM**

* + 1. **State Diagram là gì?**

Biểu đồ trạng thái (State Diagram) là dạng biểu đồ mô tả các trạng thái có thể có và sự chuyển đổi giữa các trạng thái đó khi có các sự kiện tác động của một đối tượng.

Đối với các đối tượng có nhiều trạng thái thì biểu đồ trạng thái là sự lựa chọn tốt nhất giúp chúng ta có thể hiểu rõ hơn về hệ thống.

Một cách để mô tả sự thay đổi trong một hệ thống là nói rằng các đối tượng của hệ thống thay đổi trạng thái (state) của chúng nhằm đáp ứng các sự kiện (event) và thời gian (time).

*Một vài ví dụ:*

+ Khi ta nhấn công tắc, đèn thay đổi trạng thái của nó từ “off” sang “on” hoặc ngược lại.

+ Khi ta bấm vào một điều khiển từ xa, một tivi thay đổi từ kênh này sang kênh khác.

+ Sau một khoảng thời gian thích hợp, máy giặt sẽ chuyển từ trạng thái giặt (wash) sang giũ (rinse)

* + 1. **State Diagram dùng để làm gì?**

State Diagram được sử dụng để mô tả trừu tượng về hoạt động của hệ thống. Hành vi này được phân tích và biểu diễn bằng một chuỗi các sự kiện có thể xảy ra ở một hoặc nhiều trạng thái có thể xảy ra. Bằng cách này “mỗi sơ đồ thường đại diện cho các đối tượng của một lớp duy nhất và theo dõi các trạng thái khác nhau của các đối tượng của nó thông qua hệ thống”.Biểu đồ trạng thái có thể được sử dụng để biểu diễn bằng đồ thị các máy trạng thái hữu hạn.

* + 1. **Khi nào thì sử dụng State Diagram**

Để mô hình hóa các trạng thái đối tượng của một hệ thống.

Để mô hình hóa hệ thống phản ứng. Hệ thống phản ứng bao gồm các đối tượng phản ứng.

Để xác định các sự kiện chịu trách nhiệm cho các thay đổi trạng thái.

* + 1. **Vị trí của State Diagram trong mô hình thiết kế hệ**

State Diagram thuộc ***Mô hình hóa hành vi (Behavioral Modeling)*** trong quy trình thiết kế hệ.

Mô hình hóa hành vi nhằm thể hiện hành vi bên trong hoặc một khía cạnh động của một hệ thống thông tin. Đối với State Diagram là dùng để mô tả sự xuất hiện của lớp dữ liệu bên dưới.

Chú ý rằng một state diagram khác rất nhiều so với một class diagram hoặc một use case diagram. Các diagram chúng ta đã học trước dùng để mô hình hóa hành vi của một hệ thống hoặc ít nhất cũng là một nhóm các class, object hoặc use case. Trong khi đó, state diagram lại biểu diễn các trạng thái của một object.

1. **CÁC ĐỐI TƯỢNG VẼ TRONG POWER DESIGNER**

* Trạng thái bắt đầu: (Initial State):
  + Là trạng thái khi mới được khởi tạo của object
  + Bắt buộc phải có
  + Chỉ có thể có 1 trạng thái bắt đầu

Kí hiệu:



* Trạng thái kết thúc: (Final State):
  + Chỉ vị trí kết thúc đời sống của object
  + Không nhất thiết phải thể hiện
  + Mỗi đối tượng có thể có nhiều trạng thái kết thúc

Kí hiệu:



Trong biểu đồ, đường mũi tên chỉ ra sự biến đổi từ một trạng thái sang trạng thái khác.

* Chuyển đổi (Transition)

Là sự chuyển đổi từ trạng thái này sang trạng thái khác

Đặc tả sự chuyển tiếp:

+ Sự kiện (event): cái gì đó là nguyên nhân chuyển từ trạng thái này sang trạng thái khác

+ Hầu hết chuyển tiếp đều có sự kiện. Sự kiện có thể có đối số

+ Điều kiện bảo vệ (Guard): xác định khi nào sự kiện xảy ra, thí dụ trạng thái máy bay từ Open sang Full khi chổ cuối cùng đã có người mua vé

+ Hành động (Action): hành vi không ngắt được, xảy ra như 1 phần của chuyển tiếp

Kí hiệu:



* Trạng thái đối tượng (State):

Trạng thái được xác định từ khảo sát thuộc tính lớp và quan hệ giữa các lớp

Có những sự kiện xảy ra khi đối tượng đang ở 1 trạng thái nhưng không gây ra một bước dịch chuyển mà chỉ kích hoạt một hành động hay hoạt động nào đó

Có năm loại thông tin có thể gộp trong trạng thái:

+ Hoạt động

+ Hành động vào

+ Hành động ra

+ Sự kiện

+ Lịch sử trạng thái

**Text, application, chat or text message

Description automatically generated**

* ***Hoạt động (Activity)***

• Là thao tác mà thời gian thực hiện là đáng kể

• Nó là hành vi có thể ngắt được

• Biểu diễn trong phần tử biểu đồ: do hay **/**

* ***Hành động là thao tác mà thời gian thực hiện không đáng kể***
  + ***Hành động vào (Entry)***

• Là hành vi xảy ra khi đối tượng đang chuyển vào trạng thái

• Nó là hành vi không thể ngắt được

• Biểu diễn trong phần tử biểu đồ: Entry

* + ***Hành động ra (Exit)***

• Là hành vi xảy ra khi đối tượng đang chuyển ra trạng thái

• Nó là hành vi không thể ngắt được

• Biểu diễn trong phần tử biểu đồ: Exit

Ví dụ: Trạng thái nhập mật khẩu:

Text

Description automatically generated

* ***Cách bước để vẽ sơ đồ trạng thái State Diagram***

Bước 1: Xác định trạng thái ban đầu và trạng thái kết thúc cuối cùng.

Bước 2: Xác định các trạng thái khả dĩ mà đối tượng có thể tồn tại (các giá trị biên tương ứng với các thuộc tính khác nhau hướng dẫn chúng ta xác định các trạng thái khác nhau).

Bước 3: Gắn nhãn các sự kiện kích hoạt các chuyển đổi này.

*Lưu ý:* Các quy tắc sau phải được xem xét khi vẽ biểu đồ trạng thái

+ Tên của chuyển trạng thái phải là duy nhất.

+ Tên của một trạng thái phải dễ hiểu và mô tả hành vi của một trạng thái.

+Nếu có nhiều đối tượng thì chỉ nên thực hiện các đối tượng thiết yếu.

+ Tên thích hợp cho mỗi chuyển đổi và một sự kiện phải được cung cấp.

1. **Ý NGHĨA, ĐIỀU KIỆN THIẾT KẾ STATE DIAGRAM**

UML state diagram cung cấp nhiều loại biểu tượng và chứa đựng nhiều ý tưởng nhằm mô hình hóa các thay đổi mà một object trải qua.

Trong thực tế, state diagram rất cần thiết bởi nó giúp các phân tích viên, thiết kế viên và lập trình viên hiểu được hành vi của object trong một hệ thống. Một class diagram và object diagram tương ứng chỉ cho biết khía cạnh tĩnh của hệ thống. Chúng biểu diễn sự phân cấp, mối quan hệ của các object. Chúng không cho thấy các chi tiết động của các hành vi.

Những lập trình viên cần phải biết cách thức object hoạt động bởi họ phải hiện thực những hành vi này trong phần mềm. State diagram đảm bảo rằng lập trình viên không phải “đoán mò” object sẽ hoạt động như thế nào. Với một bức tranh rõ ràng về hành vi của object, khả năng nhóm lập trình viên sẽ tạo ra một hệ thống thỏa mãn yêu cầu càng dễ trở thành hiện thực.

Một biểu đồ trạng thái thường là một sự bổ sung cho lời miêu tả một lớp. Nó chỉ ra tất cả các trạng thái mà đối tượng lớp này có thể có và những sự kiện (event) nào sẽ gây ra sự thay đổi trạng thái

Biểu đồ trạng thái không được vẽ cho tất cả các lớp, mà chỉ riêng cho những lớp có một số lượng các trạng thái được định nghĩa rõ ràng và hành vi của lớp bị ảnh hưởng và thay đổi qua các trạng thái khác nhau. Biểu đồ trạng thái cũng có thể được vẽ cho hệ thống tổng thể

Ví dụ dưới đây minh hoạt cho 1 sơ đồ trạng thái được thiết kế dựa trên điều kiện một class diagram cho trước

CLASS DIAGRAM HỆ THỐNG ATM

Diagram

Description automatically generated

STATE DIAGRAM CHO LỚP TÀI KHOẢN

**Diagram, timeline

Description automatically generated**

Mỗi sơ đồ trạng thái sẽ ánh xạ thành một thuộc tính của đối tượng

• Các trạng thái của đối tượng trong 1 sơ đồ trạng thái không giao nhau và giá trị của thuộc tính tương ứng chỉ mang 1 trong các giá trị tương ứng

• Sơ đồ trạng thái càng chi tiết sẽ phục vụ:

+ Có những xử lý thích hợp trong kiểm tra ràng buộc

+ Có những xử lý thích hợp trong xử lý biến cố

+ Phục vụ tra cứu, tìm kiếm

# **TÀI LIỆU THAM KHẢO**

* Youtube:

+ <https://www.youtube.com/watch?v=-PoMuzdUc6Y>

+ <https://www.youtube.com/watch?v=L9UCsQxuWmw>

* Blog:

+ <https://codelearn.io/sharing/5-dieu-ve-bieu-do-trang-thai-trong-uml>

+ <http://hoc-uml.blogspot.com/2017/08/bai-13-uml-statechart-diagrams.html>