ÔN TẬP MÔN PHƯƠNG PHÁP MHH

1. Mô hình hóa là gì? Hãy trình bày các mục tiêu của môn học PPMHH.

**1.1 Mô hình hóa là gì?**

Mô hình hoá là hướng nền tảng và định lượng để hiểu các hệ thống phức tạp và các hiện tượng (tự nhiên, trong cuộc sống…).

Mô hình hoá là sự bổ sung tới các phương pháp nghiên cứu cổ điển về lý thuyết và thực nghiệm.

Mô hình hoá là quá trình sản sinh ra các mô hình trừu tượng và khái niệm.

Mô hình hóa là sự cần thiết và không thể tách rời trong hoạt động khoa học.

Bản thân mô hình hóa được tạo ra bởi triết học, lý thuyết hệ thống và cả những sự mường tượng thông minh.

**1.2 Mục tiêu :**

* Cung cấp khái niệm và nguyên lý mô hình hoá
* Sử dụng các mô hình trong các biểu diễn bao gồm
  + biểu diễn dữ liệu
  + biểu diễn tri thức
  + biểu diễn hệ thống
  + biểu diễn vấn đề và lời giải
* Sử dụng một số công cụ của toán học và tin học cho mô hình hoá.
* Môn học cung cấp phương pháp hệ thống hóa để phát triễn phần mềm thông qua PPMHH hệ thống, quy trình phát triễn một phần mềm.
* Mô hình hóa sẽ tạo ra các kỹ năng và kỹ thuật nhằm tạo ra các kết quả sâu sắc, đáng tin cậy và dễ sử dụng.

Tham khảo thêm🡪

(Mô hình hóa là thay thế đối tượng các đối tượng gốc thành một mô hình để nhằm thu nhận các thông tin về đối tượng bằng cách tiến hành các thực nghiệm, tính toán trên mô hình. Lý thuyết xây dựng mô hình để hiểu biết về các đối tượng gốc gọi là lý thuyết mô hình hóa.[1]Mô hình hóa là một phương pháp khoa học để nghiên cứu các đối tượng. Nếu như các quá trình xãy ra mô hình đồng nhất - theo như các chỉ tiêu định trước – với các quá trình xãy ra bên trong đối tượng gốc thì người ta nói rằng mô hình đồng nhất với đối tượng.Lúc này, người ta có thể tiến hành các thực nghiệm trên mô hình để thu nhận thông tin về đối tượng.[1])

1. **Hãy lấy 3 ví dụ về mô hình toán học.**
2. *Mô hình của hạt trong trường vật lý điện năng*

Trong phần này, chúng ta khảo sát 1 phần tử như một hạt có khối lượng m mà nó mô tả một đường cong làm mô hình bởi hàm x:R🡪R3 được cho bởi tọa độ trong không gian như hàm thời gian. Trường điện năng được cho bởi hàm V: R3🡪R và hình chiếu là nghiệm của phương trình vi phân sau:

Lưu ý rằng mô hình này thừa nhận rằng phân tử như là khối lượng của 1 điểm mà có thể không tồn tại trong nhiều trường hợp ở đó chúng ta áp dụng mô hình như là khi chúng ta sử dụng mô hình của sự dịch chuyển của các hành tinh.

1. Trong khoa học máy tính: các mô hình kiến trúc mạng, mô hình dữ liệu, mô hình toán trong đồ họa máy tính....
2. Trong điện tử: mô hình quang phổ, mô hình năng lượng,...
3. Trong cơ học cổ điển: mô hình dao động của dây, của màng; mô hình chuyển động của tên lửa; mô hình chuyển động của tàu ngầm...
4. Trong sinh học: Mô hình về sự phát triển của [dân số](http://vi.wikipedia.org/wiki/D%C3%A2n_s%E1%BB%91). Một mô hình đơn giản cho bài toán này là [mô hình phát triển Malthus](http://vi.wikipedia.org/wiki/M%C3%B4_h%C3%ACnh_ph%C3%A1t_tri%E1%BB%83n_Malthus). Tuy nhiên, mô hình được ưa thích sử dụng lại là dùng [hàm logit](http://vi.wikipedia.org/wiki/H%C3%A0m_L%C3%B4git)
5. Trong kinh tế: Mô hình mô tả hành vi (có lí trí) của một khách hàng. Khách hàng mong muốn mua nhiều nhất các mặt hàng trong số tiền hiện có. Trong mô hình này, ta xem xét trường hợp một khách hàng phải lựa chọn để mua trong số n mặt hàng được đánh nhãn 1,2,...,n, mỗi thứ có giá là p1, p2,..., pn. Giả thiết rằng khách hàng có một hàm tiện ích U với mục đích là gán một giá trị (tương ứng cho số lượng) với mỗi mặt hàng mà khách hàng định mua x1, x2,..., xn. Mô hình còn giả thiết là khách hàng sở hữu số tiền giá trị M dùng để mua các mặt hàng và mục đích là cực đại U(x1, x2,..., xn). Bài toán cần giải quyết về mô hình hành vi của khách hàng trở thành bài toán [tối ưu hóa](http://vi.wikipedia.org/wiki/T%E1%BB%91i_%C6%B0u_h%C3%B3a_(to%C3%A1n_h%E1%BB%8Dc)), nghĩa là:

 \max U(x_1,x_2,\ldots, x_n) 

thỏa mãn:

 \sum_{i=1}^n p_i x_i \leq M.

 x_{i} \geq 0   \; \; \; \forall i \in \{1, 2, \ldots, n \} 

Mô hình này được sử dụng trong lý thuyết [cân bằng chung](http://vi.wikipedia.org/w/index.php?title=C%C3%A2n_b%E1%BA%B1ng_chung&action=edit&redlink=1), đặc biệt dùng để chứng minh sự tồn tại và tối ưu hóa Pareto của cân bằng kinh tế. Tuy nhiên, việc sử dụng mô hình này gán giá trị số để phân mức thỏa mãn của khách hàng vẫn là vấn đề tranh cãi.

1. **Nêu các nguyên lý phương pháp luận thiết kế hướng đối tượng (OOA & D). [2]**
2. Nguyên lý Mở-Đóng (Open-Closed):
   * Các thực thể phần mềm nên xây dựng theo hướng mở cho việc mở rộng và đóng đóng gói đối với việc sửa đổi.
   * Nguyên lý Mở-Đóng giúp cho việc thiết kế và xây dựng code ổn định và có thể tái sử dụng.
3. Nguyên lý thay thế Liskov (the Liskov Substitution Principle)
   * Các chức năng của hệ thống vẫn thực hiện đúng đắn nếu ta thay bất kì đối tượng nào của lớp dẫn suất bằng đối tượng của lớp kế thừa.
   * Là một phương tiện để kiểm tra xem chương trình hoặc bản thiết kế có thỏa mảng nguyên lý Mở-Đóng hay không.
4. Nguyên lý (the Single Responsibility Principle)
   * Mỗi đối tượng chỉ có một mục đích và mục đích đó được gói trong một lớp.
5. Nguyên lý nghịch đảo phụ thuộc (Dependency Inversion Principle)
   * Các đơn thể cấp cao không nên phự thuộc vào các đơn thể cấp thấp. Cả hai nên phụ thuộc vào những cái trừu tượng.
   * Cái trừu tượng không nên phụ thuộc vào cái chi tiết. Cái chi tiết nên phụ thuộc vào cái trừu tượng.
6. Nguyên lý chia tách giao diện (Interface Segregation Principle)
   * Không nên để các thực thể (phần mềm) khách phụ thuộc vào các giao diện mà chúng ta không hề sử dụng.
   * Đóng vai trò định hướng trong việc thế kế lớp trừu tượng.
7. **So sánh những đặc điểm của OOAD và OOP.**
8. **Trình bày các đặc điểm của RUP.**

* RUP là một quy trình vòng lặp phát triển phần mềm được tạo ra bởi công ty Rational Software, một bộ phận của IBM từ năm 2002 (IBM Rational).
* RUP không phải là một quy trình bó hẹp cụ thể đơn nhất nhưng là một nền tảng quy trình thích ứng với sự phát triển các tổ chức và các nhóm dự án phần mềm, tất cả sẽ chọn các yếu tố cần thiết của quy trình để phù hợp với nhu cầu, quy mô của công ty, dự án và sản phẩm.
* Unified Process được thiết kế từ đặc điểm chung, quy trình phạm vi rộng lớn và RUP là một mô tả chi tiết cụ thể.
* RUP hỗ trợ các hoạt động giữa các nhóm, phân chia công việc cho từng thành viên trong nhóm, trong từng giai đoạn khác nhau của quá trình phát triển phần mềm.
* RUP sử dụng hệ thống ký hiệu trực quan của UML và RUP được phát triển song song với UML.
* RUP là một sản phẩm tiến trình có thể tùy biến.

Vòng đời của một dự án RUP:

* Khởi tạo
* Phác thảo
* Xây dựng
* Chuyển giao

Các luồng công việc chính trong RUP:

* Mô hình hóa nghiệp vụ
* Quản lý yêu cầu
* Phân tích và thiết kế
* Cà đặt
* Kiểm thử
* Triễn khai ứng dụng
* Quản lý cấu hình và sự thay đổi
* Quản lý dự án
* Quản lý môi trường ứng dụng

1. **Trình bày những đặc điểm của Agile.**

Tuyên ngôn Agile[6]

* Cá nhân và sự tương tác hơn là quy trình và công cụ;
* Phần mềm chạy tốt hơn là tài liệu đầy đủ;
* Cộng tác với khách hàng hơn là đàm phán hợp đồng;
* Phản hồi với các thay đổi hơn là bám sát kế hoạch.

Mười hai nguyên lý[7]:

1. Thỏa mãn yêu cầu của khách hàng thông qua việc giao hàng sớm và liên tục.
2. Giao phần mềm chạy được cho khách hàng một cách thường xuyên (giao hàng tuần hơn là hàng tháng)
3. Chào đón việc thay đổi yêu cầu, thậm chí là những thay đổi yêu cầu muộn.
4. Nhà kinh doanh và kỹ sư lập trình phải làm việc cùng nhau hàng ngày trong suốt dự án
5. Các dự án được xây dựng xung quanh những cá nhân có động lực. Cung cấp cho họ môi trường và sự hỗ trợ cần thiết, và tin tưởng họ để hoàn thành công việc.
6. Trao đổi trực tiếp mặt đối mặt là phương pháp hiệu quả nhất để truyền đạt thông tin.
7. Phần mềm chạy được là thước đo chính của tiến độ.
8. Phát triển bền vững và duy trì được nhịp độ phát triển liên tục
9. Liên tục quan tâm đến kĩ thuật và thiết kế để cải tiến sự linh hoạt.
10. Sự đơn giản là cần thiết – nghệ thuật tối đa hóa lượng công việc chưa hoàn thành.
11. Nhóm tự tổ chức.
12. Thích ứng thường xuyên với sự thay đổi

## Đặc trưng Agile

Có rất nhiều phương pháp agile với các hướng tiếp cận rất khác nhau. Bên cạnh các cách thức tổ chức công việc, thiết lập quy trình, các phương pháp agile còn nghiên cứu và đưa vào sử dụng các công cụ và kĩ thuật đặc thù như công cụ tích hợp liên tục (continuous integration), kiểm thử đơn vị, mẫu thiết kế, tái cấu trúc, phát triển hướng kiểm thử (TDD), phát triển hướng hành vi (BDD), hay lập trình theo cặp v.v. để đảm bảo và gia tăng tính linh hoạt. Tuy vậy các phương pháp này chia sẻ nhiều đặc trưng giống nhau cộng tác nhóm chặt chẽ, tổ chức các nhóm tự quản, liên chức năng, tính đáp ứng cao trong suốt vòng đời của dự án.

### Tính lặp (Iterative)

### Tính tiệm tiến (Incremental) và tiến hóa (Evolutionary)

### Tính thích ứng (hay thích nghi – adaptive)

1. **Trình bày những đặc điểm của Extreme Programing.**

Lập trình cực độ (eXtreme Programming, viết tắt là XP) là một phương pháp linh hoạt dành cho các nhóm phát triển phần mềm nhỏ và trung bình xây dựng các phần mềm có yêu cầu thay đổi một cách nhanh chóng. Đối với lập trình viên, XP đảm bảo rằng họ sẽ làm những công việc hữu ích theo khả năng của họ.[8]

Mục đích cuối cùng của XP là phát triển những phần mềm với chất lượng cao nhất, với chi phí thấp nhất, ít lỗi nhất, siêu năng suất và tối đa hóa lợi nhuận đầu tư (chữ Extreme). XP làm điều này thông qua các Bốn giá trị định hướng (values), Nguyên tắc (principles) và mười hai Kĩ thuật thực hành (Practices) đặc thù theo định hướng của triết lí Agile.[9]

Các giá trị XP[9]:

* Tôn trọng (Respect)
* Giao tiếp (Communication)
* Tính đơn giản (Simplicity)
* Phản hồi (Feedback)
* Can đảm (Courage)

Các nguyên tắc XP[9]:

* Phản hồi nhanh (Rapid Feedback)
* Giả định Đơn giản (Assume Simplicity)
* Thay đổi tiệm tiến (Incremental Change)
* Sống chung với Thay đổi (Embracing Change)
* Công việc chất lượng cao (Quality Work)

1. **Tóm tắc những đặc điểm của mô hình 3-layer.**

Mô hình 3 lớp được cấu thành từ: Presentation Layers, Business Layers, và Data Layers. Các lớp này sẽ giao tiếp với nhau thông qua các dịch vụ (services) mà mỗi lớp cung cấp để tạo nên ứng dụng, lớp này cũng không cần biết bên trong lớp kia làm gì mà chỉ cần biết lớp kia cung cấp dịch vụ gì cho mình và sử dụng nó mà thôi.

Mô hình 3-layer có tính logic gồm 3 lớp :

* **Presentation Layers**

Lớp này làm nhiệm vụ giao tiếp với người dùng cuối để thu thập dữ liệu và hiển thị kết quả/dữ liệu thông qua các thành phần trong giao diện người sử dụng. Lớp này sẽ sử dụng các dịch vụ do lớp Business Logic cung cấp. Trong .NET thì có thể dùng Windows Forms, ASP.NET hay Mobile Forms để hiện thực lớp này.

Trong lớp này có 2 thành phần chính là User Interface Components và User Interface Process Components.

* **Business Logic Layer**

Lớp này thực hiện các nghiệp vụ chính của hệ thống, sử dụng các dịch vụ do lớp Data Access cung cấp, và cung cấp các dịch vụ cho lớp Presentation. Lớp này cũng có thể sử dụng các dịch vụ của các nhà cung cấp thứ 3 để thực hiện công việc của mình.

Trong lớp này có các thành phần chính là Business Components, Business Entities và Service Interface.

* **Data Layers**

Lớp này thực hiện các nghiệp vụ liên quan đến lưu trữ và truy xuất dữ liệu của ứng dụng. Thường lớp này sẽ sử dụng các dịch vụ của các hệ quản trị cơ sở dữ liệu như SQL Server, Oracle,… để thực hiện nhiệm vụ của mình. Trong lớp này có các thành phần chính là Data Access Logic, Data Sources, Servive Agents).

1. **Tóm tắc những đặc điểm của mô hình 3-tier**

* “3-tiers là một kiến trúc kiểu client/server mà trong đó giao diện người dùng (UI-user interface), các quy tắc xử lý (BR-business rule hay BL-business logic), và việc lưu trữ dữ liệu được phát triển như những module độc lập, và hầu hết là được duy trì trên các nền tảng độc lập, và mô hình 3 tầng (3-tiers) được coi là một kiến trúc phần mềm và là một mẫu thiết kế.
* 3-Tiers có tính vật lý (physical): là mô hình client-server (mỗi tier có thể đặt chung 1 nơi hoặc nhiều nơi, kết nối với nhau qua Web services, WCF, Remoting...).
* Mô hình 3-Tiers phân chia rõ rệt 3 tầng:

+ Presentation tier bao gồm các thành phần phần xử lý giao diện Graphic User Interface (GUI)  
+ Business tier gồm các thành phần Business Logic Layer (BLL), Data Access Layer (DAL) và Data Tranfer Object (DTO): xem thêm phần 3-layers  
+ Data tier lưu trữ dữ liệu, là các hệ quản trị CSDL như MS SQL Server, Oracle, SQLite, MS Access, XML files, text files,... - See more at:

1. **Nêu sự khác biệt cơ bản của mô hình 3-layer và 3-tier**

3-tiers có nghĩa là 3 tầng, 3-layer có nghĩa là 3 lớp. Về mặt ý nghĩa tầng sẽ lớn hơn lớp, mỗi tầng sẽ có nhiều lớp.

Khi dùng từ layer, chúng ta nói tới việc phân chia ứng dụng thành các thành phần một cách logic theo chức năng hoặc theo vai trò, điều này giúp phần mềm của bạn có cấu trúc sáng sủa, dễ dùng lại, từ đó giúp việc phát triển và bảo trì dễ dàng hơn. Các layer khác nhau khi được thực thi vẫn có thể nằm trong cùng một vùng bộ nhớ của một process, và hiển nhiên việc giao tiếp giữa 2 layer có thể không phải là giao tiếp giữa 2 process, đồng nghĩa với việc chúng không liên quan tới mô hình client/server.

Trái lại, tier liên quan đến cách phân chia một cách vật lý các thành phần trên các máy tính khác nhau. Điều làm nhiều người nhầm lẫn giữa layer và tier là chúng có cùng cách phân chia (presentation, business, data), tuy nhiên trên thực tế chúng khác nhau. Vì cách phân chia như trên nên 1 tier có thể chứa nhiều hơn 1 layer.

1. **Trình bày tóm tắc phương pháp luận SSADM**

SSADM là một phương pháp theo mô hình thác nước để phân tích và thiết kế hệ thống thông tin.

SSADM là một phương pháp cụ thể và xây dựng dựa trên các phương pháp khác nhau của các phương pháp phân tích và phát triển hướng cấu trúc.

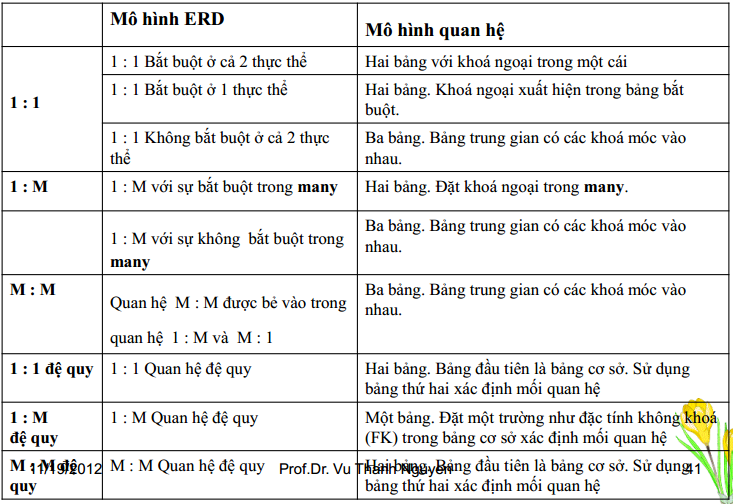
Ba kỹ thuật quan trọng nhất được sử dụng trong SSADM như sau:

* Mô hình dữ liệu logic
* Mô hình luồng dữ liệu
* Mô hình thực thể sự kiện

Các giai đoạn thực hiện:

* Phân tích hệ thống hiện tại
* Nêu cụ thể những yêu cầu hệ thống
* Lựa chọn người sử dụng của mức độ phục vụ
* Thiết kế chi tiết dữ liệu
* Thiết kế chi tiết thủ tục
* Kiểm tra thiết kế vật lý

1. **Trình bày tóm tắc phương pháp luận ETHICS**
2. **Nêu những điểm cần chú ý khi chuyển đổi gữa mô hình ERD và mô hình dữ liệu quan hệ.**



1. **Trình bày vai trò của ngôn ngữ MHH.**

Ngôn ngữ MHH là công cụ phụ vụ mô hình hóa.

Ngôn ngữ mô hình hóa dùng để trực quan hóa, đặc tả, xây dựng, viết tài liệu.

* Trực quan hóa: ngôn ngữ MHH bao gồm các tập kí hiệu phong phú để biểu diễn hệ thống đang được nghiên cứu. Nhờ có hệ thống ký hiệu chặc chẽ, nó có thể được hiểu bởi nhiều công cụ khác nhau. Nhà thiết kế, nhà lập trình khác có thể hiểu được
* Đặc tả: xây dựng các mô hình chính xác đầy đủ và không nhập nhằng.
* Xây dựng: ngôn ngữ MHH có thể kết nối với nhiều ngôn ngữ lập trình. Tức là có thể ánh xạ các mô hình sử dụng ngôn ngữ MHH về một ngôn ngữ lập trình như C++, Java, SQL,…
* Viết tài liệu: giúp xây dựng tài liệu đặc tả, tài liệu kiến trúc, tài liệu thiết kế, tài liệu kiểm.

Nó giúp tăng hiệu suất làm việc của các lập trình viên, để họ có thể giải quyết vấn đề khó khăn hơn như tính toán song song hay hệ thống phân phối.

1. **Nêu 3 ví dụ về ứng dụng của MHH.**

* Mô hình hoá khí hậu khu vực và biến đổi khí hậu là một trong những lĩnh vực đặc biệt được quan tâm không chỉ ở tầm quốc gia, khu vực mà cả trên thế giới. Nhờ sự phát triển vượt bậc và nhanh chóng của công nghệ thông tin và viễn thông, hệ thống khí hậu và các quá trình xảy ra trong nó ngày càng được con người hiểu sâu sắc hơn, đầy đủ hơn. Với sự ra đời của các công cụ, phương tiện quan trắc hiện đại, nhiều nguồn số liệu phi truyền thống đã được đưa vào sử dụng, như số liệu rada, vệ tinh, số liệu thám sát máy bay, hệ thống phao tự động trên các đại dương, v.v. Hệ thống khí hậu cũng ngày càng được mô tả chi tiết hơn, chính xác hơn bằng các mô hình toán học được giải trên các hệ thống siêu máy tính.
* Ứng dụng mô hình hóa thông tin công trình (BIM) trong đầu tư, xây dựng, quản lý và khai thác hạ tầng giao thông: Mô hình BIM là một xu hướng mới đã được áp dụng ở các nước tiên tiến như châu Âu, châu Mỹ và đang ngày càng được ứng dụng trong các ngành khác nhau. Trong đó áp dụng mạnh mẽ đầu tư quản lý và khai thác kết cấu hạ tầng GT, nhằm nâng cao hiệu quả quản lý các dự án đầu tư trong lĩnh vực xây dựng, giao thông, cơ sở hạ tầng. Ở Việt Nam. Mô hình này bước đầu đã được ứng dụng trong một số công trình về hạ tầng.
* Ứng dụng mô hình hóa trong dạy học toán: giúp học sinh tìm hiểu, khám phá các tình huống nảy sinh từ thực tiễn bằng công cụ và ngôn ngữ toán học với sự hỗ trợ của các phần mềm dạy học.

+Sử dụng mô hình hóa để tạo tình huống có vấn đề

+ Sử dụng mô hình hóa để làm sáng tỏ một số yếu tố toán học trong thực tiễn.

+ Sử dụng mô hình hóa giúp học sinh hiểu được ý nghĩa của các số liệu thống kê từ thực tiễn.

1. **Nêu các nguyên lý của mô hình hóa.**

* Tính đúng.
* Tính hội tụ.
* Tính đơn giản.
* Tính ổn định.
* Tính tối ưu ( nếu có nhiều mô hình).

1. **Nêu các tiêu chuẩn cần để đánh giá một mô hình.**

* Tính ổn định tới dữ liệu đã thử nghiệm
* Khả năng giải thích các hiện đã và sẽ xảy ra.
* Khả năng kiểm tra các các sự kiện.
* Giá thành, đặc biệt trong sự kế hợp các mô hình khác.
* Sự bác bỏ cho phép đánh giá mức độ tin tưởng.
* Thẩm mỹ.

1. **Trình bày một số nhược điểm của ngôn ngữ UML.**

* Chỉ có mã trong sự đồng bộ với mã: một viễn cảnh khác chứa đựng rằng hệ thống làm việc rất quan trọng nhưng không phải là mô hình đẹp. UML có các phương pháp cho phép biên dịch mô hình tạo ra source hoặc mã thực hiện. Tuy nhiên nó vẫn không đầy đủ rằng nó không rõ ràng do ngữ nghĩa hoạt động của UML 2.0.
* To lớn và quá phức tạp để sử dụng: Nó chứa đựng rất nhiều sơ đồ và cấu trúc mà dư thừa hoặc không thường xuyên sử dụng. Vấn đề trong học và tính kế thừa khó hiểu và không rõ về kỹ năng tiên quyết.
* Lỗi không tương xứng: như bất cứ một hệ thống ký hiệu nào, UML có khả năng biểu diễn các hệ thống xúc tích và hiệu quả hơn các hệ thống khác. Như thế người phát triển hướng về các giải pháp mà tồn tại phần giao nhau giữa khả năng của UML và ngôn ngữ thự hiện. Vấn đề một phần lỗi nếu ngôn ngữ thực hiện không bám chặt vào lý thuyết hướng đồi tượng như phần giao nhau giữa UML và ngôn ngữ thực hiện nhỏ hơn sự cần thiết.
* Sự mâu thuẫn về thẩm mỹ: luận cứ này cho rằng thiếu chú thích (hình bầu dục 2D, hộp,….) để làm cho UML có hiệu quả hơn và có trình bày đồng nhất và thẩm mỹ hơn.
* Cố gắng tất cả sự vật tới tất cả mọi người: UML là ngôn ngữ mô hình hóa có mục đích tổng quát, mà cố gắng nhận lãnh sự tích hợp với mỗi một ngôn ngữ có thể thực hiện. Trong ngữ cảnh của dự án đặc trưng, phần lớn các đặc trưng ứng dụng của UML cần phải phân định đối với người sử dụng bởi đội ngũ thiết kế hoàn thành mục tiêu cụ thể. Thêm nữa ý nghĩa khung giới hạn của UML tới một phạm vi riêng biệt thông qua hình thức mà nó không định dạng đầy đủ và tự bản thân nó là một vấn đề gây tranh cãi.

1. **Trình bày ngắn gọn về phương pháp luận SDLC.**

SDLC là phương pháp hệ thống giải quyết vấn đề nghiệp vụ.

SDLC là một cách tiếp cận có hệ thống và có trật tự để giải quyết các vấn đề liên quan đến hệ thống phần mềm hay nói cách khác, nó là một cấu trúc đối với sự phát triển của một sản phẩm phần mềm.

Phương pháp luận SDLC là phương pháp hướng dẫn, chỉ dẫn, các định hướng chỉ dẫn giúp chúng ta hoàn thành mọi hoạt động của chu trình SDLC. Nó là nền tảng của nhiều phương pháp hoạt động hay nói cách khác, từ pp SDLC, chúng ta mở rộng ra các phương pháp ứng dụng nhỏ như: mô hình thác nước, mô hình xoắn ốc,...

Nó được chia làm 7 bước.

Mỗi bước có những hoạt động duy nhất.

Ở đó thường có sự chông chéo của các bước.

Những hoạt động khác nhau có thể xuất hiện tại 1 thời điểm.

Các bước:

* Bước 1: Chỉ ra vấn đề, cơ hội và đối tượng.
* Bước 2: Xác định các yêu cầu của thông tin.
* Bước 3: Hệ thống phân tích cần thiết.
* Bước 4: Thiết kế hệ thống đề cử.
* Bước 7: Phát triển và tạo tài liệu phần mềm
* Bước 6: Thử nghiệm và bảo trì hệ thống.
* Bước 7: Vận hành và đánh giá hệ thống

Phát triển hệ thống cổ điển (SDLC) phù hợp hoàn cảnh vấn đề cấu trúc hoá tốt với vấn đề xác định tốt và yêu cầu rỏ ràng

1. **Tại sao cần tái sinh mô hình? Lấy ví dụ**

* Ontology: là sự nghiên cứu các khái niệm của thế giới thực và thiên nhiên đang hiện diện. Theo triết học ontology nghiên cứu sự sống và hiện tượng, tạo thành chủ đề cơ bản của siêu hình học. Nó tìm kiếm sự miêu tả thừa nhận các phạm trù cơ bản và mối quan hệ của sự sống hoặc hiện tượng xác định sự tồn tại và các dạng tồn tại bên trong cơ cấu tổ chức của nó. Nó là khoa học nghiên cứu cái gì, dạng nào, và cấu trúc của cấc đối tượng, đặc tính, và mối quan hệ rtong mỗi lĩnh vực của thế giới thực.
* Ontology cung cấp một bộ từ vựng chung bao gồm các khái niệm, các thuộc tính quan trọng và các định nghĩa về các khái niệm và các thuộc tính này. Ngoài bộ từ vựng, ontology còn cung cấp các ràng buộc, đôi khi các ràng buộc này được coi như các giả định cơ sở về ý nghĩa mong muốn của bộ từ vựng, nó được sử dụng trong một miền mà có thể được giao tiếp giữa người và các hệ thống ứng dụng phân tán hỗn tạp khác.
* Hiện tại, việc kết hợp các ontology là một tiến trình được làm phần lớn là thủ công, do vậy rất tốn thời gian và đắt đỏ. Việc sử dụng các ontology là cơ sở để cung cấp một định nghĩa thông dụng của các thuật ngữ cốt lõi có thể làm cho tiến trình này trở nên dễ quản lý hơn
* Ví dụ cho Ontology:

Có một thực tế là chúng ta đang ngập lụt dần trong kho dữ liệu khổng lồ mà vẫn đang phình to, lớn dần lên theo từng giây trên internet. Và trong bối cảnh như vậy thì triết lý của Google “mang lại những thông tin đáp ứng tốt nhất đối với người dùng bằng những công cụ tìm kiếm rất mạnh” như một vị cứu tinh khi nước lũ tràn về. Phải công nhận rằng triết lý ấy đã làm nên một gã khổng lồ trong công nghệ tìm kiếm thông tin trên mạng và thật khó có thể tưởng tượng nếu không có một công cụ Search như vậy thì liệu internet có phát triển được thể này.  
  
Nhưng cũng cần nhận ra rằng, gã khổng lồ này cũng không thể đáp ứng được hết nhu cầu của cư dân trên mạng. Chắc chắn bạn đã từng tìm những thông tin trên mạng bằng những công cụ Search và nhận được những kết quả không sát thực với nhu cầu tìm kiếm và đến một lúc nào đó bạn lại nhờ bạn bè hay tình cờ tìm được một trang tuyệt vời, đáp ứng đầy đủ những gì bạn cần.  
  
Vì sao vậy? Vấn đề cơ bản nằm ở cơ chế hoạt động của hệ thống web. Các trang web chứa các thông tin đa phương tiện: text, hình ảnh, âm thanh…dù ở dạng tĩnh hay động thì cũng là những thông tin chỉ có tác dụng hiển thị cho người dùng chứ không để cho máy tính hiểu và xử lý. Vì vậy, với hệ thống web như vậy thì chúng ta phải tìm kiếm thông tin trên web bằng cách nào? Đương nhiên là phải phân tích thông tin trên trang web (data mining) và công nghệ xử lý văn bản đã trở thành một để tài nghiên cứu được phát triển rất nhanh chóng. Các công cụ Search với cơ chế hoạt động dùng web spider lùng sục những trang web mới, phân tích dữ liệu trên trang web để “cố gắng” hiểu nội dung của nó, sau đó đánh chỉ số và lưu trữ các thông tin quan trọng và địa chỉ của trang web. Khi người search thông tin, họ sẽ đưa ra những từ khóa, nếu từ khóa khớp với những từ quan trọng của trang web nào đó thì trang đó sẽ là kết quả trả về của phép tìm kiếm.  
  
Nhưng cơ chế này hạn chế ở ngữ nghĩa. Vì nếu chỉ sử dụng những từ khóa để tìm kiếm thì do vấn đề ngôn ngữ (từ đa nghĩa…) mà có nhiều sự nhập nhằng và nhầm lẫn dẫn đến kết quả sai. Hơn nữa, việc hiểu nổi dung của trang web thông qua phân tích text cũng không phải là phương pháp có độ chính xác cao. Vì vậy, mà cộng đồng những nhà nghiên cứu vẫn nung nấu một mong muốn “ngữ nghĩa” hóa hệ thống web. Nghĩa là trang web bây giờ sẽ chứa đựng thông tin chủ đạo trên trang đó và biểu diễn dưới dạng mà máy tính có thể hiểu được, xử lý được. Và nếu viễn cảnh đó thành công thì triết lý “thông tin mọi nơi” sẽ biến thành “thông tin tốt nhất mọi nơi”.  
  
Vấn đề quan trọng nhất ở đây là làm thể nào để các trang web có được những thông tin ngữ nghĩa chủ đạo của nó. Ví dụ như trang web này nói về vấn đề gì? Thuộc lĩnh vực nào…Tất nhiên, câu trả lời dễ nhất sẽ là hãy để người nào cung cấp nội dung cho trang web đó viết thông tin ngữ nghĩa của web vì họ chính là người nắm rõ nhất thông tin quan trọng trên web. Vậy thì giải pháp đã có rồi???  
  
Chúng ta hãy cùng nhau tạo và một công cụ, cụ thể là một ngôn ngữ chuẩn cho web mà có thể biểu diễn được thông tin ngữ nghĩa cho web, có thể tạo một mô hình dữ liệu web mà máy tính có thể “hiểu” được. Và W3C Web Semantic Group đã cho ra đời các ngôn ngữ như RDF (Resource Desciption Framework), RDF shema, và OWL với một hi vọng như thế. Cách thức biểu diễn một cách hình thức các khái niệm nhằm mục đích tạo ngữ nghĩa cho web người ta gọi là ontology trong computer science.