# Thiết kế kiến ​​trúc

**Chủ đề được đề cập**

* Quyết định thiết kế kiến ​​trúc
* Quan điểm kiến ​​trúc
* Mẫu kiến ​​trúc
* Kiến trúc ứng dụng

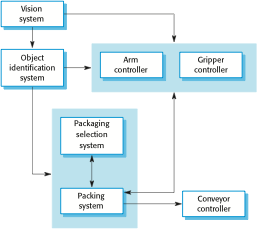
**Kiến trúc phần mềm**

* Quá trình thiết kế để xác định các hệ thống con tạo nên một hệ thống và khuôn khổ để kiểm soát và truyền thông hệ thống phụ là Thiết kế Kiến trúc.
* Đầu ra của quá trình thiết kế này là một mô tả về kiến ​​trúc phần mềm.

Thiết kế kiến ​​trúc

* Thiết kế kiến ​​trúc có liên quan với sự hiểu biết làm thế nào một hệ thống nên được tổ chức và thiết kế cấu trúc tổng thể của hệ thống đó
* Giai đoạn đầu của quá trình thiết kế hệ thống.
* Đại diện cho liên kết giữa quy trình thiết kế và đặc điểm kỹ thuật.
* Thường được thực hiện song song với một số hoạt động đặc điểm kỹ thuật.
* Nó liên quan đến việc xác định các thành phần hệ thống chính và thông tin liên lạc của chúng.

“Đầu ra của quá trình thiết kế kiến ​​trúc là một mô hình kiến ​​trúc mô tả cách hệ thống được tổ chức như một tập hợp các thành phần giao tiếp”



**Tóm tắt kiến ​​trúc**

Bạn có thể thiết kế kiến ​​trúc phần mềm ở 2 mức trừu tượng:

* Kiến trúc trong nhỏ có liên quan đến kiến ​​trúc của các chương trình riêng lẻ. Ở cấp độ này, chúng tôi quan tâm đến cách thức mà một chương trình riêng lẻ được phân tách thành các thành phần.
* Kiến trúc lớn có liên quan đến kiến ​​trúc của các hệ thống doanh nghiệp phức tạp bao gồm các hệ thống, chương trình và thành phần chương trình khác. Các hệ thống doanh nghiệp này được phân phối trên các máy tính khác nhau, có thể được sở hữu và quản lý bởi các công ty khác nhau.

Kiến trúc phần mềm là quan trọng ... Bởi vì nó ảnh hưởng đến hiệu suất, độ mạnh, khả năng phân phối và khả năng bảo trì của một hệ thống

**Ưu điểm của kiến ​​trúc rõ ràng**

Thông tin liên lạc của bên liên quan

* Kiến trúc là một bản trình bày cao cấp của hệ thống có thể được sử dụng làm trọng tâm để thảo luận bởi một loạt các bên liên quan khác nhau

Phân tích hệ thống

* Quyết định thiết kế kiến ​​trúc có ảnh hưởng sâu sắc đến việc hệ thống có thể đáp ứng các yêu cầu quan trọng như hiệu suất, độ tin cậy và khả năng bảo trì hay không.

Tái sử dụng quy mô lớn

* Mô hình kiến ​​trúc hệ thống là một mô tả nhỏ gọn, dễ quản lý về cách thức tổ chức hệ thống và cách các thành phần tương thích.
* Kiến trúc hệ thống thường giống nhau đối với các hệ thống có yêu cầu tương tự và do đó có thể hỗ trợ tái sử dụng phần mềm quy mô lớn

**Đại diện kiến ​​trúc**

* Sơ đồ khối đơn giản, không chính thức cho thấy các thực thể và các mối quan hệ là phương pháp thường được sử dụng nhất để ghi lại các kiến ​​trúc phần mềm.
* Nhưng những điều này đã bị chỉ trích vì chúng thiếu ngữ nghĩa, không hiển thị các loại mối quan hệ giữa các thực thể cũng như các đặc tính nhìn thấy được của các thực thể trong kiến ​​trúc.
* Phụ thuộc vào việc sử dụng các mô hình kiến ​​trúc. Các yêu cầu cho ngữ nghĩa mô hình phụ thuộc vào cách các mô hình được sử dụng.

**Biểu đồ hộp & đường**

* Rất trừu tượng - chúng không thể hiện bản chất của các mối quan hệ thành phần cũng như các thuộc tính hiển thị bên ngoài của các hệ thống phụ.
* Tuy nhiên, hữu ích cho giao tiếp với các bên liên quan và lập kế hoạch dự án.

**Sử dụng mô hình kiến ​​trúc**

Như một cách để tạo điều kiện thảo luận về thiết kế hệ thống

* Một quan điểm kiến ​​trúc cấp cao của một hệ thống là hữu ích cho việc giao tiếp với các bên liên quan hệ thống và lập kế hoạch dự án bởi vì nó không lộn xộn với các chi tiết. Các bên liên quan có thể liên quan đến nó và hiểu một cái nhìn trừu tượng của hệ thống. Sau đó, họ có thể thảo luận về toàn bộ hệ thống mà không bị nhầm lẫn chi tiết.

Như một cách để viết tư liệu một kiến ​​trúc đã được thiết kế

* Mục tiêu ở đây là tạo ra một mô hình hệ thống hoàn chỉnh cho thấy các thành phần khác nhau trong một hệ thống, các giao diện và các kết nối của chúng.

**Quyết định thiết kế kiến ​​trúc**

* Thiết kế kiến ​​trúc là một quá trình sáng tạo nơi bạn thiết kế một tổ chức hệ thống sẽ đáp ứng các yêu cầu chức năng và phi chức năng của một hệ thống
* Bởi vì nó là một quá trình sáng tạo, các hoạt động trong quá trình này phụ thuộc vào loại hệ thống đang được phát triển, nền tảng và kinh nghiệm của kiến ​​trúc sư hệ thống và các yêu cầu cụ thể cho hệ thống.
* Do đó, hữu ích khi nghĩ đến thiết kế kiến ​​trúc là một loạt các quyết định được đưa ra chứ không phải là một chuỗi các hoạt động
* Tuy nhiên, một số quyết định phổ biến kéo dài tất cả các quy trình thiết kế và các quyết định này ảnh hưởng đến các đặc điểm phi chức năng của hệ thống.
* Trong quá trình thiết kế kiến ​​trúc, các kiến ​​trúc sư hệ thống phải thực hiện một số quyết định về cấu trúc ảnh hưởng sâu sắc đến hệ thống và quá trình phát triển của nó.

***Dựa trên kiến ​​thức và kinh nghiệm của họ, họ phải xem xét những câu hỏi cơ bản sau đây về hệ thống….***

* Có kiến ​​trúc ứng dụng chung nào có thể được sử dụng không?
* Hệ thống sẽ được phân phối như thế nào?
* Kiểu kiến ​​trúc nào phù hợp?
* Cách tiếp cận nào sẽ được sử dụng để cấu trúc hệ thống?
* Làm thế nào hệ thống sẽ được phân tích thành các mô-đun?
* Chiến lược kiểm soát nào nên được sử dụng?
* Thiết kế kiến ​​trúc sẽ được đánh giá như thế nào?
* Kiến trúc nên được ghi chép như thế nào?

**Kiến trúc tái sử dụng**

* Các hệ thống trong cùng một miền thường có kiến ​​trúc tương tự phản ánh khái niệm miền.
* Các dòng sản phẩm ứng dụng được xây dựng xung quanh một kiến ​​trúc cốt lõi với các biến thể đáp ứng các yêu cầu cụ thể của khách hàng.
* Kiến trúc của một hệ thống có thể được thiết kế xung quanh một trong nhiều kiểu kiến ​​trúc hoặc "kiểu" hơn.
  + Chúng nắm bắt được bản chất của một kiến ​​trúc và có thể được khởi tạo theo nhiều cách khác nhau.

**Kiến trúc & đặc điểm hệ thống**

Do mối quan hệ chặt chẽ giữa các yêu cầu phi chức năng và kiến ​​trúc phần mềm, kiểu kiến ​​trúc và cấu trúc cụ thể mà bạn chọn cho một hệ thống phải phụ thuộc vào các yêu cầu hệ thống phi chức năng

Hiệu suất

* Bản địa hóa các hoạt động quan trọng và giảm thiểu thông tin liên lạc. Sử dụng các thành phần lớn hơn là hạt mịn.

Bảo vệ

* Sử dụng kiến ​​trúc lớp với các tài sản quan trọng trong các lớp bên trong.

An toàn

* Bản địa hóa các tính năng quan trọng về an toàn trong một số ít các hệ thống phụ.

khả dụng

* Bao gồm các thành phần dự phòng và cơ chế cho khả năng chịu lỗi.

Duy trì

* Sử dụng hạt mịn, thành phần thay thế.

**1. Hiệu suất**

* Nếu hiệu suất là một yêu cầu quan trọng, kiến ​​trúc phải được thiết kế để bản địa hóa các hoạt động quan trọng trong một số lượng nhỏ các thành phần, với các thành phần này được triển khai trên cùng một máy tính thay vì được phân phối trên mạng.
* Điều này có thể có nghĩa là sử dụng một vài thành phần tương đối lớn hơn là các thành phần nhỏ, hạt mịn, làm giảm số lượng các thành phần truyền thông. Bạn cũng có thể xem xét các tổ chức hệ thống thời gian chạy cho phép hệ thống được sao chép và thực hiện trên các bộ vi xử lý khác nhau

**2. Bảo mật**

Nếu bảo mật là một yêu cầu quan trọng, nên sử dụng cấu trúc phân lớp cho kiến ​​trúc, với các tài sản quan trọng nhất được bảo vệ trong các lớp trong cùng, với mức độ xác thực bảo mật cao được áp dụng cho các lớp này

**3. An toàn**

* Nếu an toàn là một yêu cầu quan trọng, kiến ​​trúc phải được thiết kế sao cho tất cả các hoạt động liên quan đến an toàn đều nằm trong một thành phần đơn lẻ hoặc trong một số lượng nhỏ các thành phần.
* Điều này làm giảm chi phí và các vấn đề về xác nhận an toàn và làm cho nó có thể cung cấp các hệ thống bảo vệ liên quan có thể tắt hệ thống một cách an toàn trong trường hợp thất bại.

**Quan điểm kiến ​​trúc**

* Quan điểm hoặc quan điểm nào hữu ích khi thiết kế và ghi lại kiến ​​trúc của hệ thống?
* Ký hiệu nào nên được sử dụng để mô tả các mô hình kiến ​​trúc?
* Mỗi mô hình kiến ​​trúc chỉ hiển thị một khung nhìn hoặc phối cảnh của hệ thống.
  + Nó có thể cho thấy cách một hệ thống bị phân tách thành các mô-đun, cách các quy trình thời gian chạy tương tác hoặc các cách khác nhau trong đó các thành phần hệ thống được phân phối trên một mạng. Đối với cả thiết kế và tài liệu, bạn thường cần trình bày nhiều quan điểm của kiến ​​trúc phần mềm.

**4 + 1 mô hình xem kiến ​​trúc phần mềm**

* Một khung nhìn lô-gic, cho thấy các trừu tượng chính trong hệ thống như các đối tượng hoặc các lớp đối tượng.
* Một khung nhìn quy trình, trong đó cho thấy, tại thời gian chạy, hệ thống bao gồm các quá trình tương tác.
* Một khung nhìn phát triển, cho thấy cách phần mềm bị phân hủy để phát triển.
* Một khung nhìn vật lý, hiển thị phần cứng hệ thống và cách các thành phần phần mềm được phân phối trên các bộ xử lý trong hệ thống.
* Liên quan đến các trường hợp sử dụng hoặc kịch bản (+1)

**Mẫu kiến ​​trúc**

* Các mẫu là một phương tiện đại diện, chia sẻ và sử dụng lại kiến ​​thức.
* Một mô hình kiến ​​trúc là một mô tả cách điệu về thực hành thiết kế tốt, đã được thử nghiệm và thử nghiệm trong các môi trường khác nhau.
* Các mẫu nên bao gồm thông tin về thời điểm chúng và khi nào chúng không hữu ích.
* Các mẫu có thể được biểu diễn bằng cách sử dụng các mô tả dạng bảng và đồ họa.

**Mẫu Model-View-Controller (MVC)**

Sự miêu tả

* Phân tách bản trình bày và tương tác từ dữ liệu hệ thống. Hệ thống được cấu trúc thành ba thành phần logic tương tác với nhau. Thành phần Model quản lý dữ liệu hệ thống và các hoạt động liên quan trên dữ liệu đó. Thành phần Chế độ xem xác định và quản lý cách dữ liệu được hiển thị cho người dùng. Thành phần Bộ điều khiển quản lý tương tác của người dùng (ví dụ: các lần nhấn phím, các lần nhấp chuột, v.v.) và chuyển các tương tác này đến Chế độ xem và Mô hình.

Thí dụ

* Hình 6.4 cho thấy kiến ​​trúc của một hệ thống ứng dụng dựa trên web được tổ chức bằng cách sử dụng mẫu MVC.

Khi đã sử dụng

* Được sử dụng khi có nhiều cách để xem và tương tác với dữ liệu. Cũng được sử dụng khi các yêu cầu tương lai cho tương tác và trình bày dữ liệu chưa được biết.

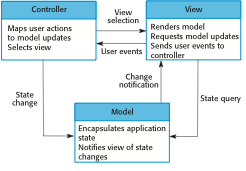
Ưu điểm

* Cho phép dữ liệu thay đổi độc lập với biểu diễn của nó và ngược lại. Hỗ trợ trình bày cùng một dữ liệu theo các cách khác nhau với các thay đổi được thực hiện trong một biểu diễn được hiển thị trong tất cả chúng.

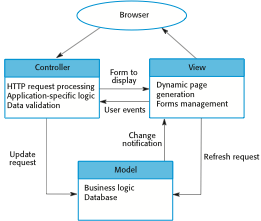
Nhược điểm

* Có thể liên quan đến mã phức tạp và mã bổ sung khi mô hình dữ liệu và tương tác rất đơn giản.

**Tổ chức Model-View-Controller**



**Kiến trúc ứng dụng web sử dụng mẫu MVC**



**Kiến trúc lớp**

* Được sử dụng để mô hình hóa giao diện của các hệ thống phụ.
* Tổ chức hệ thống thành một tập hợp các lớp (hoặc các máy trừu tượng), mỗi hệ thống cung cấp một tập hợp các dịch vụ.
* Hỗ trợ sự phát triển gia tăng của các hệ thống con trong các lớp khác nhau. Khi một giao diện lớp thay đổi, chỉ có lớp liền kề bị ảnh hưởng.
* Tuy nhiên, thường là nhân tạo để cấu trúc hệ thống theo cách này.

**Mô hình kiến ​​trúc Layered**

Sự miêu tả

* Tổ chức hệ thống thành các lớp với chức năng liên quan được liên kết với mỗi lớp. Một lớp cung cấp dịch vụ cho lớp ở trên nó để các lớp mức thấp nhất đại diện cho các dịch vụ cốt lõi có khả năng được sử dụng trong toàn bộ hệ thống.

Thí dụ

* Một mô hình phân lớp của một hệ thống để chia sẻ các tài liệu bản quyền được lưu giữ trong các thư viện khác nhau, như trong Hình 6.7.

Khi đã sử dụng

* Được sử dụng khi xây dựng các cơ sở mới trên các hệ thống hiện có; khi sự phát triển được lan truyền qua nhiều nhóm với trách nhiệm của mỗi nhóm đối với một lớp chức năng; khi có yêu cầu về bảo mật đa cấp.

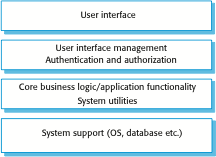
Ưu điểm

* Cho phép thay thế toàn bộ các lớp, miễn là giao diện được duy trì. Các cơ sở dự phòng (ví dụ: xác thực) có thể được cung cấp trong mỗi lớp để tăng tính tin cậy của hệ thống.

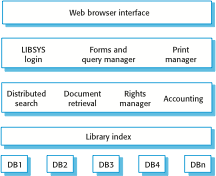
Nhược điểm

* Trong thực tế, việc cung cấp sự tách biệt giữa các lớp thường khó khăn và một lớp cao cấp có thể phải tương tác trực tiếp với các lớp cấp thấp hơn là thông qua lớp ngay bên dưới nó. Hiệu suất có thể là một vấn đề vì nhiều mức độ giải thích của một yêu cầu dịch vụ vì nó được xử lý ở mỗi lớp.

**Một kiến ​​trúc lớp chung**



**Kiến trúc của hệ thống LIBSYS**



**Những điểm chính**

* Kiến trúc phần mềm là mô tả về cách thức tổ chức hệ thống phần mềm.
* Quyết định thiết kế kiến ​​trúc bao gồm các quyết định về loại ứng dụng, sự phân bố của hệ thống, kiểu kiến ​​trúc được sử dụng.
* Các kiến ​​trúc có thể được ghi lại từ một số quan điểm hoặc quan điểm khác nhau như một khung nhìn khái niệm, một khung nhìn logic, một khung nhìn quy trình và một khung nhìn phát triển.
* Các mẫu kiến ​​trúc là một phương tiện tái sử dụng kiến ​​thức về các kiến ​​trúc hệ thống chung. Chúng mô tả kiến ​​trúc, giải thích khi nào nó có thể được sử dụng và mô tả những ưu điểm và nhược điểm của nó.

**Kiến trúc kho lưu trữ**

* Các hệ thống phụ phải trao đổi dữ liệu. Điều này có thể được thực hiện theo hai cách:
  + Dữ liệu được chia sẻ được lưu trữ trong một cơ sở dữ liệu hoặc kho lưu trữ trung tâm và có thể được truy cập bởi tất cả các hệ thống con;
  + Mỗi hệ thống con duy trì cơ sở dữ liệu riêng của mình và truyền dữ liệu một cách rõ ràng cho các hệ thống con khác.
* Khi lượng lớn dữ liệu được chia sẻ, mô hình chia sẻ lưu trữ thường được sử dụng nhất, đây là cơ chế chia sẻ dữ liệu hiệu quả.

**Mẫu Repository**

Sự miêu tả

* Tất cả dữ liệu trong một hệ thống được quản lý trong một kho lưu trữ trung tâm có thể truy cập được tới tất cả các thành phần hệ thống. Các thành phần không tương tác trực tiếp, chỉ thông qua kho lưu trữ.

Thí dụ

* Hình 6.9 là một ví dụ về một IDE nơi các thành phần sử dụng một kho lưu trữ thông tin thiết kế hệ thống. Mỗi công cụ phần mềm tạo ra thông tin mà sau đó có sẵn để sử dụng bởi các công cụ khác.

Khi đã sử dụng

* Bạn nên sử dụng mẫu này khi bạn có một hệ thống trong đó khối lượng lớn thông tin được tạo ra phải được lưu trữ trong một thời gian dài. Bạn cũng có thể sử dụng nó trong các hệ thống theo hướng dữ liệu trong đó việc đưa dữ liệu vào kho lưu trữ kích hoạt một hành động hoặc công cụ.

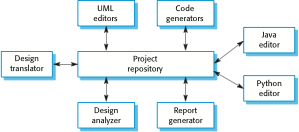
Ưu điểm

* Các thành phần có thể độc lập - chúng không cần biết về sự tồn tại của các thành phần khác. Các thay đổi được thực hiện bởi một thành phần có thể được truyền cho tất cả các thành phần. Tất cả dữ liệu có thể được quản lý một cách nhất quán (ví dụ: các bản sao lưu được thực hiện cùng một lúc) vì tất cả dữ liệu đều ở cùng một nơi.

Nhược điểm

* Kho lưu trữ là một điểm lỗi duy nhất nên các vấn đề trong kho lưu trữ ảnh hưởng đến toàn bộ hệ thống. Có thể không hiệu quả trong việc tổ chức tất cả các giao tiếp thông qua kho lưu trữ. Việc phân phối kho lưu trữ trên nhiều máy tính có thể khó khăn.

**Kiến trúc kho lưu trữ cho một IDE**



**Kiến trúc máy khách-máy chủ**

* Mô hình hệ thống phân tán cho thấy cách dữ liệu và xử lý được phân phối trên một loạt các thành phần.
* Có thể được triển khai trên một máy tính.
* Đặt các máy chủ độc lập cung cấp các dịch vụ cụ thể như in, quản lý dữ liệu, v.v.
* Đặt các khách hàng gọi đến các dịch vụ này.
* Mạng cho phép khách hàng truy cập máy chủ.

**Mẫu Client-server**

Description

* Trong kiến ​​trúc client-server, chức năng của hệ thống được tổ chức thành các dịch vụ, với mỗi dịch vụ được phân phối từ một máy chủ riêng biệt. Khách hàng là người dùng của các dịch vụ này và truy cập máy chủ để sử dụng chúng.

When used

* Được sử dụng khi dữ liệu trong cơ sở dữ liệu được chia sẻ phải được truy cập từ một loạt các vị trí. Bởi vì máy chủ có thể được nhân rộng, cũng có thể được sử dụng khi tải trên hệ thống là biến.

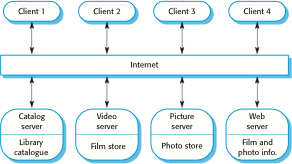
Advantages

* Ưu điểm chính của mô hình này là các máy chủ có thể được phân phối trên mạng. Chức năng chung (ví dụ: dịch vụ in) có thể có sẵn cho tất cả khách hàng và không cần phải được thực hiện bởi tất cả các dịch vụ.

Disadvantages

* Mỗi dịch vụ là một điểm thất bại nên dễ bị tấn công từ chối dịch vụ hoặc lỗi máy chủ. Hiệu suất có thể không thể đoán trước được vì nó phụ thuộc vào mạng cũng như hệ thống. Có thể là vấn đề quản lý nếu máy chủ được sở hữu bởi các tổ chức khác nhau.

**Kiến trúc máy khách-máy chủ cho thư viện phim**



**Kiến trúc ống & lọc**

* Chuyển đổi chức năng xử lý đầu vào của chúng để tạo ra kết quả đầu ra.
* Có thể được gọi là mô hình đường ống và bộ lọc (như trong vỏ UNIX).
* Các biến thể của phương pháp này rất phổ biến. Khi các biến đổi là tuần tự, đây là một mô hình tuần tự theo lô được sử dụng rộng rãi trong các hệ thống xử lý dữ liệu.
* Không thực sự phù hợp với các hệ thống tương tác.

**Mẫu ống và bộ lọc**

Description

* Việc xử lý dữ liệu trong một hệ thống được tổ chức sao cho mỗi thành phần xử lý (bộ lọc) là rời rạc và thực hiện một loại chuyển đổi dữ liệu. Các luồng dữ liệu (như trong một đường ống) từ một thành phần này đến thành phần khác để xử lý.

When used

* Thường được sử dụng trong các ứng dụng xử lý dữ liệu (cả hàng loạt và dựa trên giao dịch), trong đó các đầu vào được xử lý trong các giai đoạn riêng biệt để tạo ra các kết quả đầu ra liên quan.

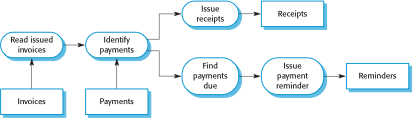
Advantages

* Dễ hiểu và hỗ trợ tái sử dụng chuyển đổi. Kiểu luồng công việc khớp với cấu trúc của nhiều quy trình nghiệp vụ. Sự tiến hóa bằng cách thêm các phép biến đổi là đơn giản. Có thể được thực hiện như một hệ thống tuần tự hoặc đồng thời.

Disadvantages

* Định dạng truyền dữ liệu phải được thỏa thuận giữa các phép chuyển đổi truyền thông. Mỗi phép biến đổi phải phân tích cú pháp đầu vào của nó và tách đầu ra của nó thành dạng đã đồng ý. Điều này làm tăng chi phí của hệ thống và có thể có nghĩa là không thể tái sử dụng các phép biến đổi chức năng sử dụng các cấu trúc dữ liệu không tương thích.

**Ví dụ về cấu trúc đường ống và bộ lọc**



**Kiến trúc ứng dụng**

* Các hệ thống ứng dụng được thiết kế để đáp ứng nhu cầu của tổ chức.
* Khi các doanh nghiệp có nhiều điểm chung, các hệ thống ứng dụng của họ cũng có xu hướng có một kiến ​​trúc chung phản ánh các yêu cầu ứng dụng.
* Kiến trúc ứng dụng chung là một kiến ​​trúc cho một loại hệ thống phần mềm có thể được cấu hình và điều chỉnh để tạo ra một hệ thống đáp ứng các yêu cầu cụ thể.

**Sử dụng kiến ​​trúc ứng dụng**

* Là điểm khởi đầu cho thiết kế kiến ​​trúc.
* Là một danh sách kiểm tra thiết kế. Như một cách để tổ chức công việc của nhóm phát triển.
* Là một phương tiện đánh giá các thành phần để tái sử dụng.
* Là một từ vựng để nói về các loại ứng dụng.

**Ví dụ về các loại ứng dụng**

Ứng dụng xử lý dữ liệu

* Các ứng dụng điều khiển dữ liệu xử lý dữ liệu theo lô mà không có sự can thiệp của người dùng rõ ràng trong quá trình xử lý.

Ứng dụng xử lý giao dịch

* Các ứng dụng tập trung vào dữ liệu xử lý các yêu cầu của người dùng và cập nhật thông tin trong cơ sở dữ liệu hệ thống.

Hệ thống xử lý sự kiện

* Các ứng dụng mà các hành động hệ thống phụ thuộc vào việc diễn giải sự kiện từ môi trường của hệ thống.

Hệ thống xử lý ngôn ngữ

* Các ứng dụng mà ý định của người dùng được chỉ định bằng ngôn ngữ chính thức được xử lý và diễn giải bởi hệ thống.

**Ví dụ về loại ứng dụng**

* Tập trung ở đây là về xử lý giao dịch và hệ thống xử lý ngôn ngữ.
* Hệ thống xử lý giao dịch
  + Hệ thống thương mại điện tử;
  + Hệ thống đặt chỗ.
* Hệ thống xử lý ngôn ngữ
  + Trình biên dịch;
  + Trình thông dịch lệnh.

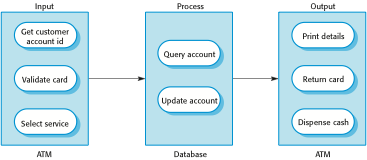
**Hệ thống xử lý giao dịch**

* Xử lý các yêu cầu của người dùng về thông tin từ cơ sở dữ liệu hoặc yêu cầu cập nhật cơ sở dữ liệu.
* Từ góc độ người dùng, giao dịch là:
  + Bất kỳ chuỗi hoạt động chặt chẽ nào thỏa mãn mục tiêu;
  + Ví dụ: - tìm thời gian chuyến bay từ Luân Đôn đến Paris.
* Người dùng thực hiện các yêu cầu không đồng bộ cho dịch vụ mà sau đó được xử lý bởi người quản lý giao dịch.

**Cấu trúc của các ứng dụng xử lý giao dịch**

6.14 TransactionProcSys.eps

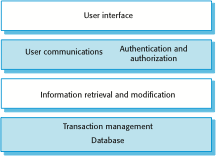
**Kiến trúc phần mềm của một hệ thống ATM**



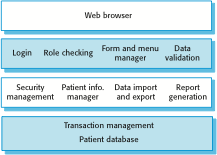
**Kiến trúc hệ thống thông tin**

* Các hệ thống thông tin có một kiến ​​trúc chung có thể được tổ chức như một kiến ​​trúc phân lớp.
* Đây là những hệ thống dựa trên giao dịch khi tương tác với các hệ thống này thường liên quan đến các giao dịch cơ sở dữ liệu.
* Các lớp bao gồm:
  + Giao diện người dùng
  + Liên lạc của người dùng
  + Thu hồi thông tin
  + Cơ sở dữ liệu hệ thống

**Kiến trúc hệ thống thông tin lớp**



**Kiến trúc của MHC-PMS**



**Hệ thống thông tin dựa trên web**

* Thông tin và các hệ thống quản lý tài nguyên hiện nay thường là các hệ thống dựa trên web, nơi các giao diện người dùng được thực hiện bằng cách sử dụng một trình duyệt web.
* Ví dụ, các hệ thống thương mại điện tử là các hệ thống quản lý tài nguyên dựa trên Internet chấp nhận các đơn đặt hàng điện tử cho hàng hóa hoặc dịch vụ và sau đó sắp xếp việc phân phối các hàng hoá hoặc dịch vụ này cho khách hàng.
* Trong hệ thống thương mại điện tử, lớp ứng dụng cụ thể bao gồm chức năng bổ sung hỗ trợ ‘giỏ hàng’ trong đó người dùng có thể đặt một số mặt hàng trong các giao dịch riêng biệt, sau đó thanh toán cho tất cả chúng trong cùng một giao dịch.

**Triển khai máy chủ**

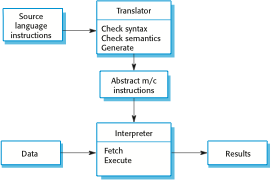
Các hệ thống này thường được triển khai dưới dạng máy chủ / kiến ​​trúc khách hàng nhiều tầng (được thảo luận trong Chương 18)

* Máy chủ web chịu trách nhiệm về tất cả các liên lạc của người dùng, với giao diện người dùng được triển khai bằng trình duyệt web;
* Máy chủ ứng dụng chịu trách nhiệm triển khai logic ứng dụng cụ thể cũng như các yêu cầu lưu trữ và truy xuất thông tin;
* Máy chủ cơ sở dữ liệu di chuyển thông tin đến và từ cơ sở dữ liệu và xử lý quản lý giao dịch.

**Hệ thống xử lý ngôn ngữ**

* Chấp nhận một ngôn ngữ tự nhiên hoặc nhân tạo làm đầu vào và tạo ra một số đại diện khác của ngôn ngữ đó.
* Có thể bao gồm một thông dịch viên để thực hiện các hướng dẫn bằng ngôn ngữ đang được xử lý.
* Được sử dụng trong các tình huống mà cách dễ nhất để giải quyết vấn đề là mô tả thuật toán hoặc mô tả dữ liệu hệ thống
  + Các công cụ Meta-case mô tả công cụ xử lý, các quy tắc phương thức, vv và tạo ra các công cụ.

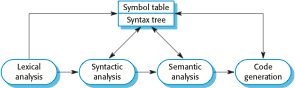
**Kiến trúc của một hệ thống xử lý ngôn ngữ**



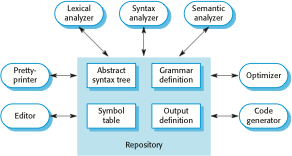
**Thành phần trình biên dịch**

* Một máy phân tích từ vựng, có các mã thông báo ngôn ngữ đầu vào và chuyển đổi chúng thành một biểu mẫu bên trong.
* Một bảng biểu tượng, chứa thông tin về tên của các thực thể (các biến, tên lớp, tên đối tượng, vv) được sử dụng trong văn bản đang được dịch.
* Trình phân tích cú pháp, kiểm tra cú pháp của ngôn ngữ đang được dịch.
* Một cây cú pháp, là một cấu trúc bên trong biểu diễn chương trình đang được biên dịch.
* Một trình phân tích ngữ nghĩa sử dụng thông tin từ cây cú pháp và bảng biểu tượng để kiểm tra tính chính xác ngữ nghĩa của văn bản ngôn ngữ nhập liệu.
* Một trình tạo mã mà ‘đi’ cây cú pháp và tạo ra mã máy trừu tượng.

**Kiến trúc trình biên dịch ống và bộ lọc**



**Kiến trúc kho lưu trữ cho hệ thống xử lý ngôn ngữ**



**Key points**

* Các mô hình kiến ​​trúc hệ thống ứng dụng giúp chúng ta hiểu và so sánh các ứng dụng, xác nhận các thiết kế hệ thống ứng dụng và đánh giá các thành phần có quy mô lớn để tái sử dụng.
* Hệ thống xử lý giao dịch là các hệ thống tương tác cho phép thông tin trong cơ sở dữ liệu được một số người dùng truy cập và sửa đổi từ xa.
* Hệ thống xử lý ngôn ngữ được sử dụng để dịch văn bản từ ngôn ngữ này sang ngôn ngữ khác và thực hiện các hướng dẫn được chỉ định bằng ngôn ngữ nhập liệu. Chúng bao gồm một dịch giả và một máy trừu tượng thực thi ngôn ngữ được tạo ra.