

Chương 1 – GIỚI THIỆU CHUNG

1.1. Tương tác người – máy là gì

1.2. Đặc điểm tâm sinh lý con người trong quá trình giao tiếp

1.3. Phương tiện giao tiếp của máy tính

Chương 1 – GIỚI THIỆU CHUNG

1.1. Tương tác người – máy là gì

- *Khái niệm chung*
- *Những chuyên ngành liên quan đến HCI*
- *Mục đích của việc thiết kế giao diện người dùng tốt*
- *Tính tiện dụng của một hệ thống*
- *Đối tượng môn học*

Khái niệm chung

- *Tương tác người – máy* (Human Computer Interaction – HCI): là việc nghiên cứu con người (người dùng), công nghệ máy tính và tác động qua lại giữa các đối tượng đó.
- *Mục đích của việc nghiên cứu HCI*: phát triển hay cải thiện tính an toàn, tính tiện dụng, tính hiệu quả của hệ thống; tạo ra hệ thống dùng được và an toàn.
- *Các thành phần mà HCI nghiên cứu*:
 - *Hình thức*: các hình thức giao tiếp giữa người và máy.
 - *Chức năng*: các chức năng mới trong giao tiếp giữa người và máy.
 - *Cài đặt*: cài đặt các giao diện trên thiết bị.

Những chuyên ngành liên quan đến HCI

- *Tâm lý học, xã hội học, triết học*: hiểu được sự cảm nhận thông tin, quá trình nhận thức, kỹ năng giải quyết vấn đề.
- *Sinh lý học, công thái học*: hiểu được khả năng vật lý của con người.
- *Khoa học máy tính và công nghệ phần mềm*: xây dựng các phần mềm cần thiết.
- *Thiết kế đồ họa, thiết kế âm thanh, hình ảnh*: thiết kế các giao diện một cách hiệu quả.
- ...

Mục đích của việc thiết kế giao diện người dùng tốt (1)

** Về mặt lập trình:*

- Thiết kế giao diện tốt sẽ cho phép giảm thời gian lập trình cho sản phẩm.
- Nếu thiết kế giao diện sai sẽ phải mất thời gian thiết kế lại.
- Nếu thiết kế giao diện không tốt, cũng phải thiết kế lại. Nếu không sửa chữa được, người sử dụng sẽ phải dùng giao diện không tốt.

Mục đích của việc thiết kế giao diện người dùng tốt (2)

** Về mặt kinh tế:*

- Giảm chi phí đào tạo
- Giảm những lỗi người dùng
- Tăng năng suất lao động
- Tạo ra những sản phẩm có chất lượng cao
- Tăng khả năng bán được của sản phẩm

Mục đích của việc thiết kế giao diện người dùng tốt (3)

* *Về mặt an toàn:*

- Giảm những bệnh nghề nghiệp

- Giảm những lỗi nguy hiểm đến tính mạng

Tính tiện dụng của một hệ thống – 5 tiêu chí của Nielson (1)

** Tính dễ học:*

- Các hệ thống tương tác phải dễ học.
- Thể hiện qua thời gian và công sức bỏ ra để đạt được một trình độ sử dụng nhất định.

** Tính hiệu quả:*

- Một hệ thống tương tác tốt phải có tính hiệu quả.
- Được đánh giá thông qua: mức hiệu suất công việc đạt được; thời gian hoàn thành công việc ở mức cao nhất; tần suất lỗi.

Tính tiện dụng của một hệ thống – 5 tiêu chí của Nielson (2)

** Tính dễ nhớ:*

- Thể hiện qua giao diện thiết kế hợp lý, thân thiện với người sử dụng.
- Hệ thống tương tác được thiết kế có tính dễ nhớ sẽ khiến người sử dụng dễ học, dễ dàng sử dụng.

** Tính dự đoán lỗi:*

- Người dùng thường dự đoán kết quả của một sự tương tác dựa vào những kiến thức mà họ thu được từ những lần tương tác trước.
- Hệ thống nên hỗ trợ các suy luận hay dự đoán này bằng cách luôn luôn đưa ra những thông tin phản hồi nhất quán.

Tính tiện dụng của một hệ thống – 5 tiêu chí của Nielson (3)

** Đáp ứng tính chủ quan:*

- Là khả năng đáp ứng của một hệ thống đối với những người dùng khác nhau trong những trường hợp khác nhau.
- Đánh giá đáp ứng tính chủ quan thông qua hiệu suất và số lỗi tạo ra trong các tình huống khác nhau.

Đối tượng môn học

- *Con người*: nghiên cứu đặc điểm tâm sinh lý của con người trong quá trình giao tiếp.
- *Máy tính*: nghiên cứu các phương tiện giao tiếp của máy tính.
- *Mô hình tương tác và các dạng tương tác*: các kỹ thuật giao tiếp từ truyền thống đến hiện đại.
- *Thiết kế tương tác người – máy*: quy trình thiết kế, các chuẩn trong thiết kế, các mô hình người dùng,...
- *Mô hình hệ thống*: các phương pháp biểu diễn đối thoại và ứng dụng; các kỹ thuật phân tích nhiệm vụ.
- *Đánh giá hệ thống*: các kỹ thuật đánh giá giao tiếp người dùng, đánh giá sản phẩm.

Chương 1 – GIỚI THIỆU CHUNG

1.1. Tương tác người – máy là gì

1.2. Đặc điểm tâm sinh lý con người trong quá trình giao tiếp

1.3. Phương tiện giao tiếp của máy tính

Chương 1 – GIỚI THIỆU CHUNG

1.2. Đặc điểm tâm sinh lý con người trong quá trình giao tiếp

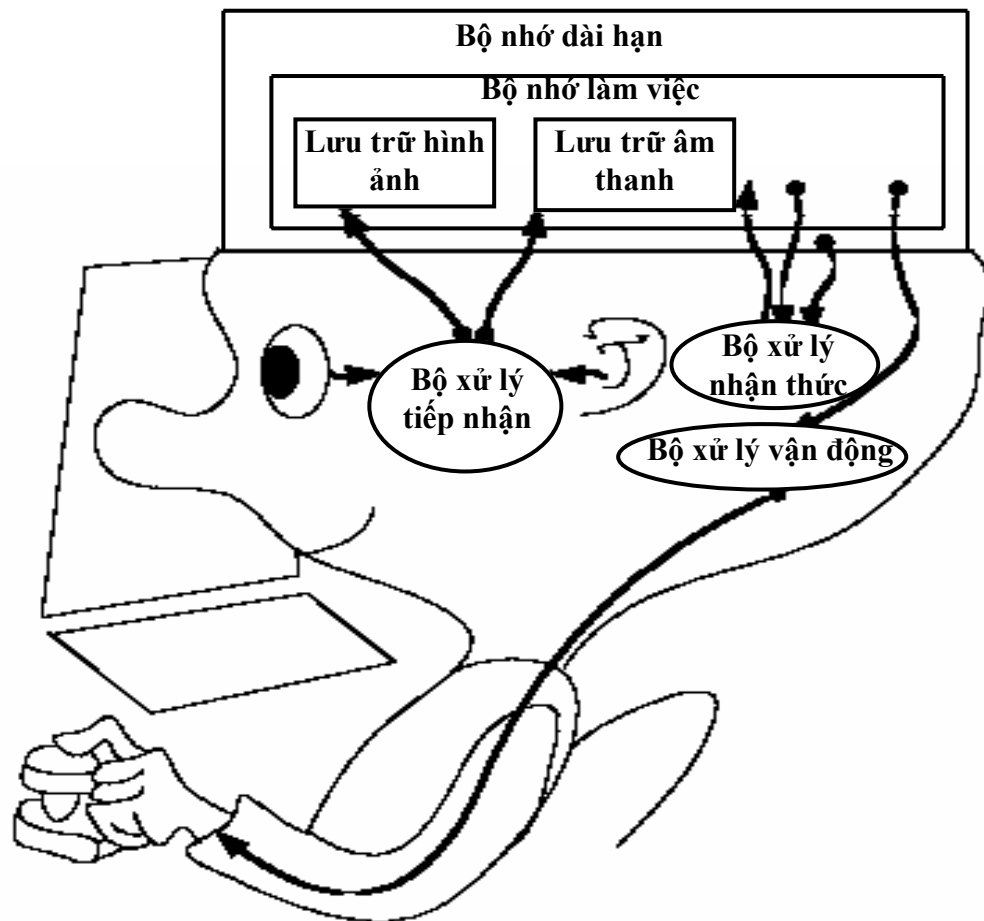
1.2.1. Mô hình đơn giản về bộ xử lý của con người

1.2.2. Các kênh vào – ra thông tin của con người

1.2.3. Trí nhớ con người và ảnh hưởng tới quá trình giao tiếp

1.2.4. Suy diễn và giải quyết vấn đề

1.2.1. Mô hình đơn giản về bộ xử lý của con người

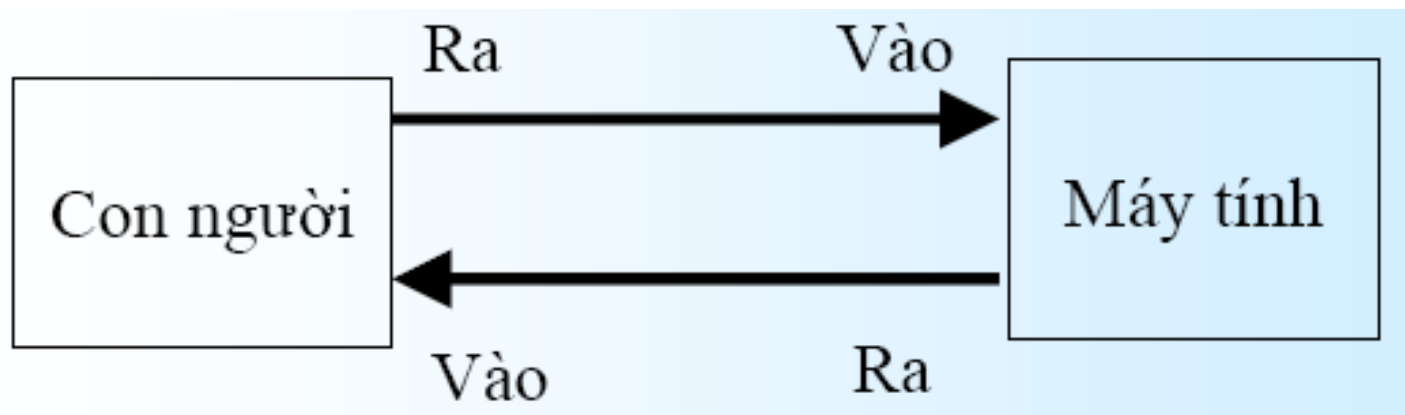


Mô hình của Card, Moral và Newell (1983):

- *Hệ thống cảm nhận (Perceptual System)*
- *Hệ thống nhận thức (Cognitive System)*
- *Hệ thống xử lý (Motor System)*

1.2.2. Các kênh vào – ra thông tin của con người

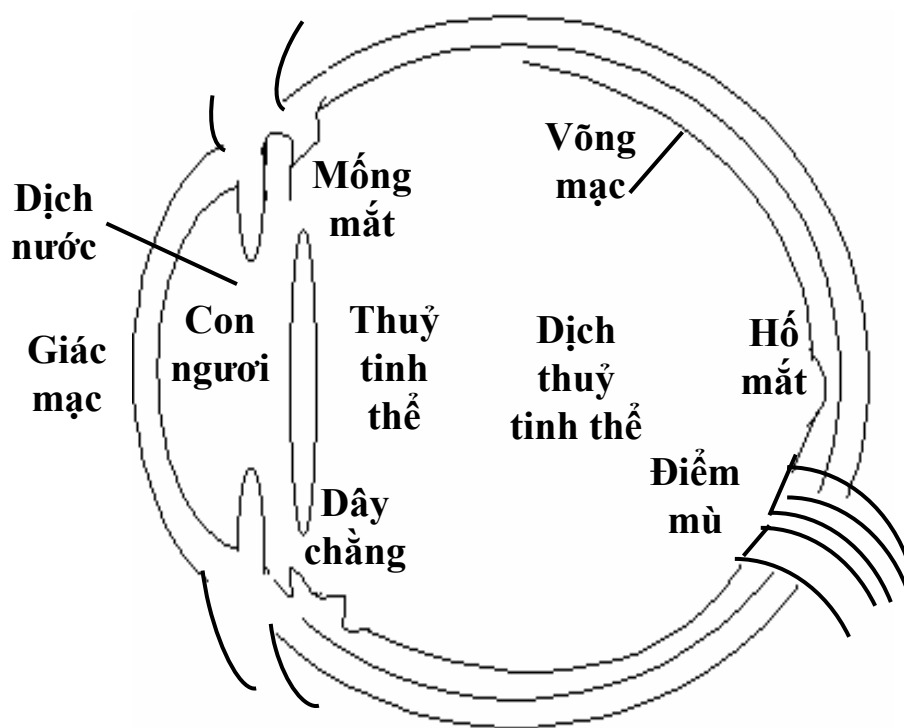
- Đầu vào của con người chủ yếu xuất hiện thông qua các giác quan. Đầu ra xuất hiện thông qua sự điều khiển vận động của các cơ quan phản ứng kích thích.
- Có 5 giác quan chính: thị giác, thính giác, xúc giác, vị giác và khứu giác.
- Các cơ quan phản ứng kích thích có rất nhiều, bao gồm: chân, tay, các ngón tay, mắt, đầu và hệ thống phát âm.



1.2.2.1. Thị giác (1)

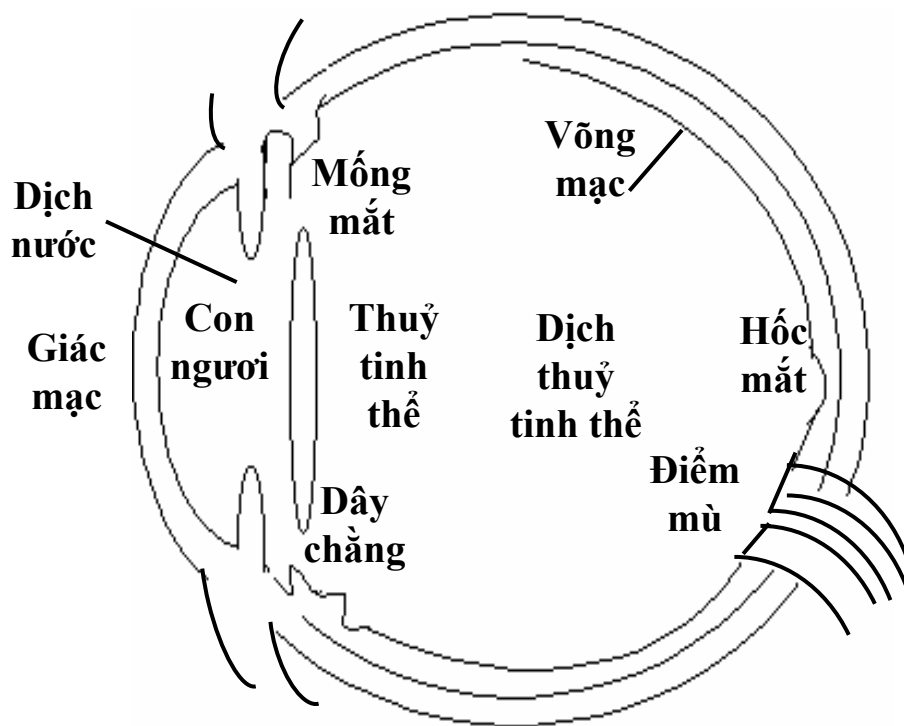
- Đối với một người bình thường, quan sát bằng mắt là nguồn tiếp nhận thông tin chủ yếu.
- Quá trình tiếp nhận bằng thị giác có thể được chia thành 2 giai đoạn:
 - + Giai đoạn cảm nhận: nhận các kích thích vật lý từ thế giới bên ngoài.
 - + Giai đoạn xử lý, giải nghĩa các kích thích: các tính chất vật lý của các kích thích mắt người nhận được sẽ được phân tích theo kích thước, màu sắc, độ sáng, độ tương phản.

1.2.2.1. Thị giác (2) – Cấu tạo mắt người



- Mắt tiếp nhận ánh sáng và biến đổi thành năng lượng điện, chuyển tới não.
- Giác mạc và thủy tinh thể ở phía trước mắt hội tụ ánh sáng thành một hình ảnh sắc nét nằm ở phía đuôi mắt, võng mạc.
- Võng mạc rất nhạy sáng và nó chứa hai loại tế bào tiếp nhận ánh sáng: tế bào hình que và tế bào hình nón.

1.2.2.1. Thị giác (3) – Cấu tạo mắt người

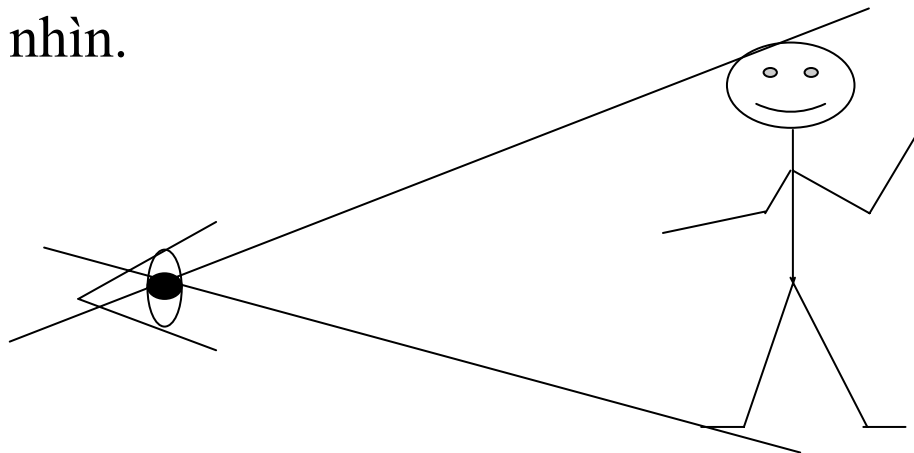


- Tế bào hình que là tế bào cực kỳ nhạy sáng. Mỗi mắt có khoảng 120 triệu tế bào hình que chủ yếu nằm ở các viền của võng mạc.
- Các tế bào hình nón không nhạy sáng bằng các tế bào hình que. Có 3 loại tế bào hình nón, mỗi loại nhạy cảm với một bước sóng ánh sáng khác nhau: màu đỏ, màu lục và màu lam. Mắt có khoảng 6 triệu tế bào hình nón, chủ yếu tập trung ở hốc mắt.
- Điểm mù: nơi nối các dây thần kinh thị giác với mắt. Điểm mù không có tế bào hình nón hoặc tế bào hình que.

1.2.2.1. Thị giác (4) – Thu nhận bằng thị giác

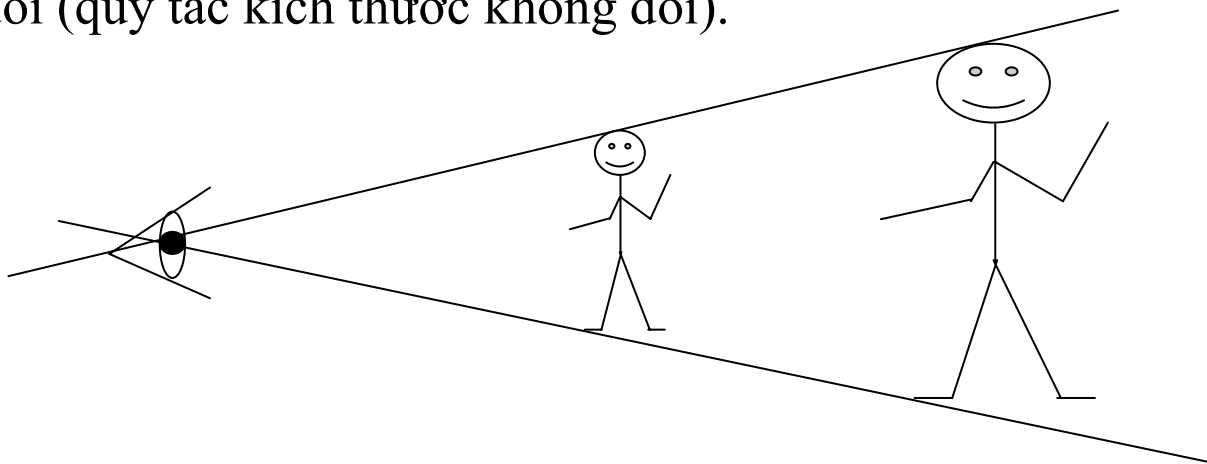
* *Cảm nhận về kích thước, khoảng cách:*

- Sự cảm nhận chính xác phụ thuộc vào kích thước đối tượng và khoảng cách từ đối tượng đến mắt.
- Ánh sáng được phản chiếu từ đối tượng tạo ra một ảnh ảo ngược chiều trên võng mạc. Kích thước của hình ảnh đó được đặc trưng bởi góc nhìn.
- Góc nhìn là góc giới hạn bởi hai đường thẳng từ đỉnh và từ chân đối tượng đi qua tâm nhìn.



1.2.2.1. Thị giác (5) – Thu nhận bằng thị giác

- Góc nhìn xác định phạm vi quan sát được của đối tượng là bao nhiêu. Góc nhìn thường được đo bằng độ, phút hoặc giây.
- Góc nhìn phụ thuộc vào kích thước đối tượng và khoảng cách từ đối tượng đến mắt.
- Ảnh hưởng của góc nhìn đến sự cảm nhận của con người về kích thước:
 - + Nếu góc nhìn quá nhỏ: không cảm nhận được đối tượng.
 - + Sự cảm nhận về kích thước đối tượng là một hằng số, ngay cả khi góc nhìn thay đổi (quy tắc kích thước không đổi).



1.2.2.1. Thị giác (6) – Thu nhận bằng thị giác

* *Cảm nhận độ sáng tối:*

- Độ sáng tối là đáp ứng chủ quan của mức độ sáng.
- Độ sáng tối phụ thuộc vào số tia sáng rơi trên bề mặt đối tượng và tính chất phản xạ của bề mặt.
- Có thể đo độ sáng tối bằng quang kế.
- *Độ tương phản:* là độ nổi của ảnh đối tượng so với nền.
- Độ sáng tối giúp ta phân biệt sự khác nhau về mức sáng. Khi ánh sáng tối, sẽ khó nhìn đối tượng hơn.
- Tính sắc bén của thị giác tăng khi độ sáng tăng. Tuy nhiên, khi độ sáng tăng thì sự lập lờ cũng tăng.

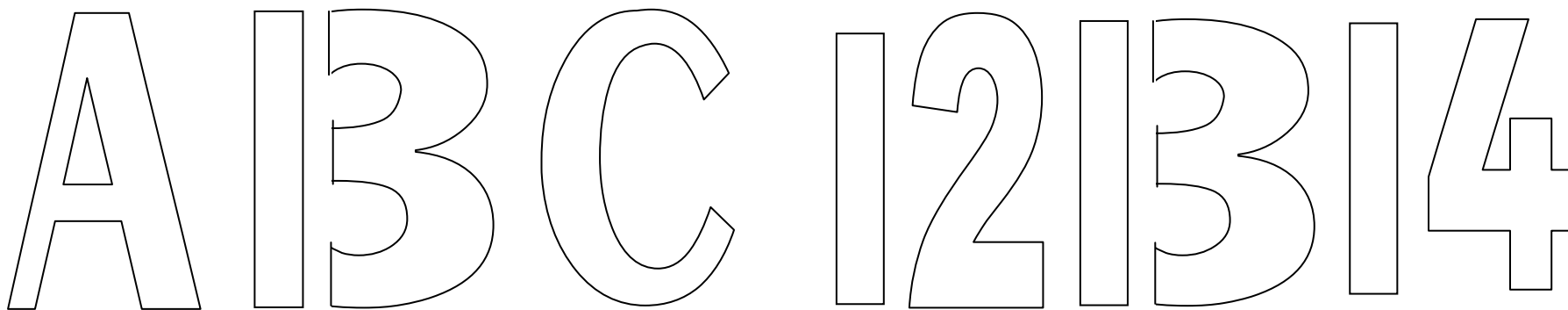
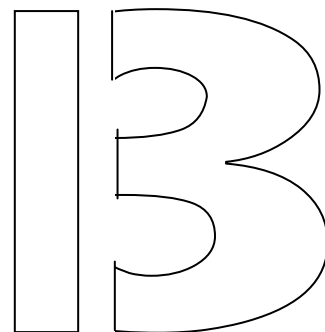
1.2.2.1. Thị giác (7) – Thu nhận bằng thị giác

* *Cảm nhận màu sắc:*

- *Sắc thái màu:* xác định bởi bước sóng của ánh sáng. Màu xanh lam (xanh da trời) có bước sóng ngắn; màu xanh lục (xanh lá cây) có bước sóng trung bình và màu đỏ có bước sóng dài.
- *Cường độ màu:* là độ sáng của màu sắc.
- *Độ bão hòa:* là tổng số lượng màu trắng có trong màu. Khi tăng thêm lượng ánh sáng trắng, độ bão hòa sẽ thay đổi.
- Trung bình mắt người có thể phân biệt được khoảng 150 màu. Khi thay đổi cường độ và độ bão hòa, mắt người có thể cảm nhận tới hàng triệu màu.
- Hiện tượng mù màu: là không có khả năng cảm nhận màu sắc. Có khoảng 8% đàn ông và 1% phụ nữ bị mù màu.

1.2.2.1. Thị giác (8) – Khả năng của hệ thống thị giác

* Cảm nhận phụ thuộc vào ngữ cảnh:



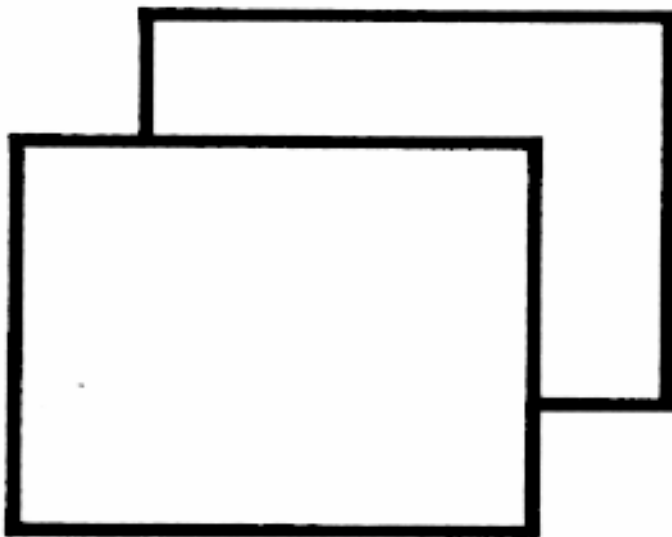
1.2.2.1. Thị giác (9) – Khả năng của hệ thống thị giác

* *Cảm nhận hình ảnh ẩn:*

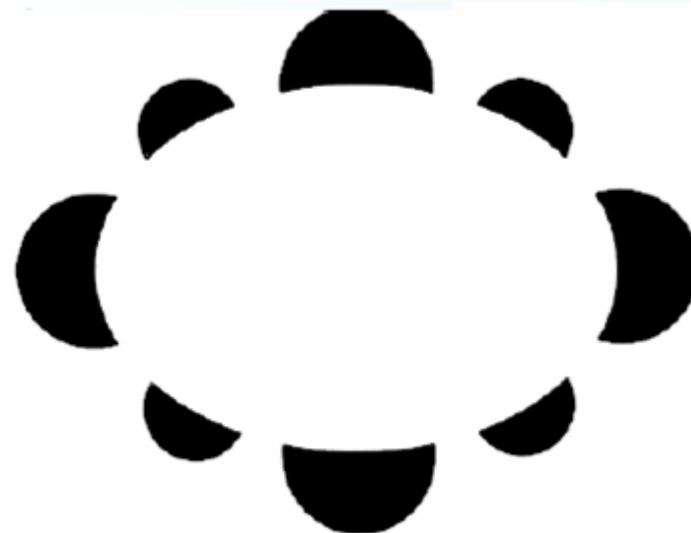


1.2.2.1. Thị giác (10) – Khả năng của hệ thống thị giác

c

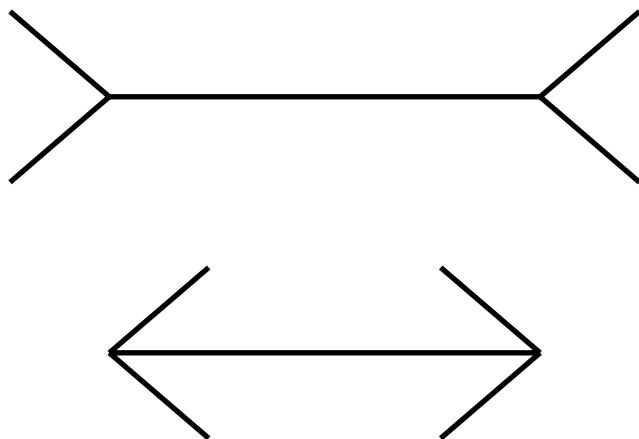


d

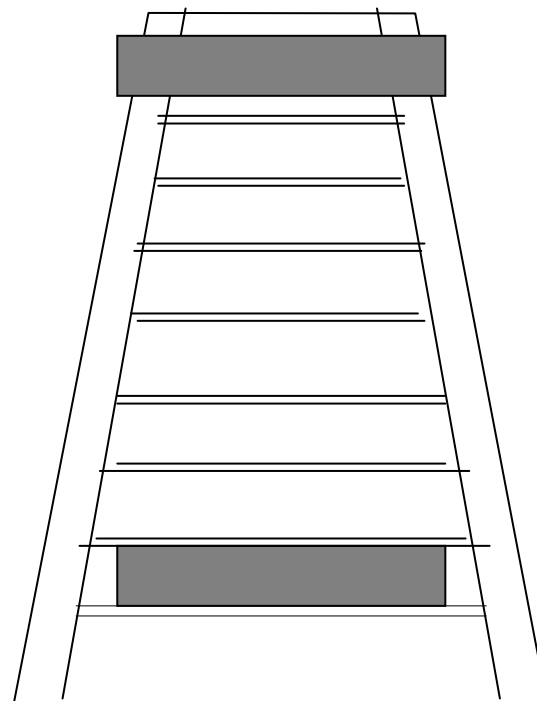


1.2.2.1. Thị giác (11) – Hạn chế: hiện tượng ảo giác

* Ảo giác quang học:



* Ảo giác Ponzo:



1.2.2.1. Thị giác (12) – Cảm nhận và xử lý văn bản

- Các giai đoạn của xử lý đọc:
 - + Thu nhận mẫu từ trang giấy.
 - + Giải mã mẫu: đối chiếu mẫu với các mẫu cơ bản của ngôn ngữ.
 - + Phân tích cú pháp và ngữ nghĩa, phân tích các cụm từ hoặc các câu.
- Trong khi đọc, mắt thực hiện các chuyển động lên xuống. Đoạn văn bản càng phức tạp, sự chuyển động của mắt càng nhiều.
- Tốc độ đọc của một người bình thường: 200 từ/phút.
- Font chữ tiêu chuẩn từ 9 đến 12 dễ đọc tỉ lệ với độ giãn cách dòng. Chiều dài dòng cũng ảnh hưởng đến tính dễ đọc.
- Đọc từ màn hình máy tính thường chậm hơn đọc trên giấy.
- Chữ đen trên nền trắng (độ tương phản âm) dễ đọc hơn chữ trắng trên nền đen (độ tương phản dương).

1.2.2.2. Thính giác (1) – Âm thanh

- Âm thanh là sự thay đổi hay rung động khi không khí bị nén.
- Âm thanh được đặc trưng bởi các yếu tố: tần số; độ vang và âm sắc.
 - Tần số âm thanh là một hằng số.
 - Độ vang phụ thuộc vào độ khuếch đại.
 - Âm sắc là phẩm chất đặc trưng cho âm thanh phát ra.
- Thính giác bắt đầu với những dao động trong không khí hoặc hoặc các sóng âm thanh.
- Tai tiếp nhận những dao động đó và truyền chúng, qua các giai đoạn khác nhau, đến các dây thần kinh thính giác.

1.2.2.2. Thính giác (2) – Cấu tạo tai người

- Tai ngoài là phần nhìn thấy được của tai. Gồm 2 phần: loa tai, là cấu trúc được gắn vào 2 bên đầu, và ống thính giác. Tai ngoài bảo vệ phần tai giữa để bị tổn thương khỏi nguy hiểm. Đồng thời, loa tai và ống thính giác còn dùng để khuếch đại âm thanh.
- Tai giữa là một khoang nhỏ nối với tai ngoài qua màng nhĩ và với tai trong qua ốc tai. Bên trong khoang là các xương nhỏ nhất trong cơ thể con người. Sóng âm thanh đi dọc theo ống thính giác và làm rung màng nhĩ, làm cho các xương nhỏ dao động theo, truyền các dao động đến ốc tai và đi vào tai trong.
- Bên trong ốc tai là các tế bào rất nhỏ, gọi là các lông mao. Lông mao sẽ bị cong đi do sự dao động trong chất dịch ốc tai và phát ra một tín hiệu hoá học để tạo ra các xung thần kinh thính giác.

1.2.2.2. Thính giác (3) – Cảm nhận âm thanh

- Tai người có thể nghe được âm thanh với tần số từ khoảng 20 Hz đến 15 kHz.
- Tai có thể phân biệt được các thay đổi tần số trong phạm vi nhỏ hơn 1.5 Hz đối với các tần số thấp, và sẽ phân biệt ít chính xác hơn với các tần số cao.
- Hệ thống thính giác thực hiện chức năng lọc đối với các âm thanh nhận được, cho phép chúng ta bỏ qua phần tín hiệu nhiễu mà chỉ tập trung vào các thông tin quan trọng.
- Nếu âm thanh quá ồn, hoặc tần số của nó quá nhỏ, chúng ta sẽ không có khả năng phân biệt được các âm thanh khác nhau.

1.2.2.2. Thính giác (4) – Sử dụng thính giác trong HCI

- Âm thanh ít được sử dụng trong khi thiết kế giao diện, mà thường chỉ giới hạn trong các âm thanh cảnh báo:
 - + Thông báo khi gõ nhầm nút.
 - + Thông báo khi vào một chương trình.
 - + Thông báo khi máy sắp hết pin.
 - ...
- Hiện nay, âm thanh đang được nghiên cứu theo các hướng:
 - + Tổng hợp tiếng nói: để nghe đọc tài liệu thay vì nhìn tài liệu, nhằm phục vụ cho người khiếm thị.
 - + Dùng âm nhạc để tạo ra các hiệu ứng trong trình diễn nội dung.

1.2.2.3. Xúc giác (1)

- Xúc giác cho chúng ta những thông tin cần thiết về môi trường quanh ta. Nó thông báo cảm giác mà chúng ta sẽ có được khi cầm nắm vào một vật và do đó nó đóng vai trò như một cảnh báo.
- Xúc giác cũng cung cấp thông tin phản hồi, khi chúng ta vận động cơ.
- Xúc giác là phương tiện quan trọng của phản hồi và thực tế là trong các hệ thống máy tính việc sử dụng các thông tin phản hồi là tương đối nhiều.
- Đối với những người mà các giác quan khác như thị giác hoặc thính giác bị hỏng, thì xúc giác sẽ trở nên vô cùng quan trọng.

1.2.2.3. Xúc giác (2)

- Xúc giác nhận kích thích thông qua da.
- Da có 3 kiểu cảm nhận:
 - + Cảm nhận nhiệt: phản ứng lại với nóng và lạnh.
 - + Cảm nhận sức căng do áp suất: phản ứng lại với sức ép căng thẳng, sự nóng giận và đau đớn.
 - + Cảm nhận cơ: phản ứng lại với áp lực.
- HCI chỉ liên quan đến cảm nhận cơ.

1.2.2.3. Xúc giác (3)

- Cảm nhận cơ được chia làm hai dạng:
 - + Dạng đáp ứng nhanh: phản ứng lại với áp lực ngay lập tức sau khi da nhận kích thích. Khi áp lực càng gia tăng thì chúng phản ứng lại càng nhanh. Tuy nhiên, chúng sẽ ngừng phản ứng nếu áp lực tác động lên là áp lực liên tục.
 - + Dạng đáp ứng chậm: phản ứng với các áp lực tác động một cách liên tục.
- Có những vùng cơ thể có độ nhạy cảm hoặc có tính nhạy bén cao hơn những vùng khác.
- Độ nhạy cảm của các ngón tay là lớn nhất.

1.2.3. Trí nhớ con người và ảnh hưởng tới quá trình giao tiếp

- Mô hình bộ xử lý thông tin con người của Card, Moral và Newell: bao gồm ba hệ thống con, mỗi hệ thống có bộ xử lý và bộ nhớ riêng.
- Ba kiểu bộ nhớ:
 - Bộ nhớ cảm nhận (sensory memory)
 - Bộ nhớ ngắn hạn, hay bộ nhớ làm việc (Short term memory)
 - Bộ nhớ dài hạn (Long term memory)



1.2.3.1. Bộ nhớ cảm nhận

- Chứa các kích thích nhận được từ các giác quan như: nghe, nhìn, sờ. Ứng với mỗi giác quan có một bộ nhớ cảm nhận riêng. Tại bộ nhớ cảm nhận, các kích thích được mã hóa.
- Thông tin trong bộ nhớ cảm nhận được lưu theo cách viết đè: thông tin mới sẽ thay thế các thông tin cũ.
- Thông tin từ bộ nhớ cảm nhận qua bộ lọc rồi chuyển đến bộ nhớ ngắn hạn.
- Khi thông tin trong bộ nhớ ngắn hạn bão hòa, thông tin không được chuyển sang nữa.
- Thông tin được lưu trong bộ nhớ cảm nhận với thời gian khác nhau. Với bộ nhớ thị giác, thời gian tồn tại là khoảng 200ms, với bộ nhớ thính giác là khoảng 1500ms.

1.2.3.2. Bộ nhớ ngắn hạn (1)

- Tổ chức như kiểu thanh ghi của máy tính, là bộ nhớ hoạt động của bộ xử lý trung tâm.
- Ba đặc trưng chính của bộ nhớ ngắn hạn:
 - + Thời gian truy nhập nhanh, khoảng 70ms.
 - + Thông tin hư hỏng nhanh, khoảng 200ms.
 - + Dung lượng bộ nhớ hạn chế.
- Thông thường, một người bình thường nhớ được khoảng 7 ± 2 đoạn: dãy các con số, các sự kiện, ...
- Khi bộ nhớ ngắn hạn bão hòa, sự kích hoạt của đoạn mới sẽ xóa đi thông tin trong bộ nhớ nếu nó không được nhắc lại.

1.2.3.2. Bộ nhớ ngắn hạn (2)

- Khi các đoạn được hình thành từ các mẫu tốt, khả năng nhớ sẽ tăng lên.
 - Đưa ra một số điện thoại không theo thứ tự: 0904505559
Số điện thoại trên được phân thành các đoạn: 0904–50–55– 59
 - Xét xâu chữ sau: “ thecatrunupthetree”
Chia xâu chữ trên thành các đoạn 2 hay 3 chữ cái: “the cat run up the tree”
- Ký tự cuối cùng trong một câu hay sự kiện cuối cùng trong một dãy là dễ nhớ nhất.

1.2.3.3. Bộ nhớ dài hạn (1)

- Có vai trò như bộ nhớ trung tâm và bộ nhớ thứ cấp của máy tính.
- Lưu trữ thông tin, thông tin đó có thể được đọc hoặc bị thay đổi.
- Các đặc trưng chính của bộ nhớ dài hạn:
 - + Bộ nhớ dài hạn có cấu trúc tuyến tính và có dung lượng không hạn chế.
 - + Thời gian truy cập tương đối chậm, khoảng 1/10s.
 - + Thời gian hư hỏng chậm.

1.2.3.3. Bộ nhớ dài hạn (2) – Tổ chức bộ nhớ

- Bộ nhớ rời rạc (bộ nhớ đoạn – episode memory):
 - Dùng để biểu diễn sự ghi nhớ của chúng ta về các sự kiện và các kinh nghiệm theo một chuỗi liên tục.
 - Bộ nhớ rời rạc giúp chúng ta có thể nhớ lại các sự kiện thực tế đã xảy ra tại một thời điểm nào đó trong cuộc đời.
- Bộ nhớ ngữ nghĩa (semantic memory):
 - Chứa đựng các mẫu tin có cấu trúc và các sự kiện, các khái niệm và các kỹ năng mà chúng ta thu thập được.
 - Thông tin trong bộ nhớ ngữ nghĩa được kế thừa từ những kinh nghiệm trong bộ nhớ rời rạc, những kinh nghiệm này giúp cho chúng ta có thể học được các khái niệm hoặc các sự việc mới.

1.2.3.3. Bộ nhớ dài hạn (3) – Quá trình xử lý thông tin

- Lưu trữ/nhớ lại thông tin:

- + Thông tin từ bộ nhớ ngắn hạn chuyển đến bộ nhớ dài hạn theo cơ chế “tổng duyệt”, nghĩa là do quá trình nhắc đi nhắc lại từ bộ nhớ ngắn hạn.
- + Lượng thông tin nhớ được tỷ lệ với thời gian học (kết luận của Ebbinghaus – 1885).
- + Việc ghi nhớ hay học tập sẽ đạt hiệu quả tốt hơn nếu nó được dàn trải đều theo thời gian (kết luận của Baddeley và Longman – 1978).
- + Các từ mô tả đối tượng dễ nhớ hơn các từ mô tả khái niệm.
Ví dụ: các từ “Nhà”, “Cây” dễ nhớ hơn các từ “Chủ nghĩa” , “Quá khứ”.
- + Những thông tin có ý nghĩa và quen thuộc thì dễ nhớ hơn.
Ví dụ: đọc thần thoại Hy Lạp khó nhớ hơn đọc truyền thuyết Việt Nam.

1.2.3.3. Bộ nhớ dài hạn (4) – Quá trình xử lý thông tin

- Quên thông tin:

- + Nguyên nhân: thông tin bị hư hỏng hoặc có hiện tượng giao thoa.
- + Thuyết thông tin bị hư hỏng: theo Ebbinghaus, thông tin trong bộ nhớ sẽ dần dần bị mất đi theo thang logarit, nghĩa là thông tin mới sẽ hư hỏng nhanh hơn thông tin cũ.
- + Thuyết giao thoa: sự tiếp nhận thông tin mới sẽ gây nên sự mất mát thông tin cũ. Các thông tin cũ đôi khi cũng có thể can thiệp lại thông tin mới.
- + Ảnh hưởng của cảm xúc: các sự kiện có nhiều cảm xúc sẽ ít bị quên hơn các sự kiện có ít cảm xúc.

1.2.3.3. Bộ nhớ dài hạn (5) – Quá trình xử lý thông tin

- Truy cập thông tin:

- + Kiểu gọi lại: thông tin được tạo lại (sao chép lại) từ bộ nhớ.
- + Kiểu nhận dạng: so sánh thông tin với các thông tin đã có trong bộ nhớ.
- + Quá trình nhận dạng đơn giản hơn quá trình gọi lại, vì có thông tin làm gợi ý.

Ví dụ: nhớ ra mặt một người khó hơn nhận ra người đó;
câu hỏi lựa chọn dễ hơn câu hỏi thông thường.

1.2.4. Suy diễn và giải quyết vấn đề

- Suy diễn và giải quyết vấn đề là quá trình xử lý và quản lý thông tin.
- Đây là quá trình phức tạp nhất phân biệt con người với các hệ thống xử lý thông tin khác.
- Con người có khả năng sử dụng thông tin để lập luận, giải quyết vấn đề và tiến hành các hoạt động khi thông tin có tính cục bộ hoặc không đầy đủ.
- Lập luận là quá trình con người sử dụng tri thức đã có để đưa ra kết luận hay suy diễn điều mới về lĩnh vực quan tâm.
- Giải quyết vấn đề là quá trình tìm lời giải cho một nhiệm vụ chưa biết bằng các tri thức chúng ta có.

1.2.4.1. Lập luận (1) – Suy luận

- Đưa ra các kết luận một cách logic từ các giả thiết.
- Các lập luận kiểu suy luận thường được biểu diễn dưới dạng: “Nếu ... thì ...”

Ví dụ: Nếu là chiều thứ 5 thì chúng ta có tiết HCl.

Hôm nay là thứ 5.

Suy ra : chúng ta có tiết HCl.

- Tuy nhiên, đôi khi các lập luận kiểu suy luận mang đến những kết luận sai.

Ví dụ: Một số sinh viên đã đi làm.

Một số người đi làm là lãnh đạo.

Suy ra : một số sinh viên là lãnh đạo.

1.2.4.1. Lập luận (2) – Qui nạp

- Là quá trình suy diễn từ cái đã biết sang cái chưa biết.
- Đây là cách lập luận không tin cậy hoàn toàn.

Ví dụ: Mọi con voi chúng ta gặp đều có vòi.

Kết luận: mọi con voi đều có vòi.

- Có thể không chứng minh được là đúng, chỉ có thể tìm càng nhiều bằng chứng càng tốt để hỗ trợ cho ý kiến.
- Để chứng minh cách lập luận này sai, chỉ cần đưa ra một ví dụ sai.

1.2.4.1. Lập luận (3) – Phản chứng

- Đưa ra cách giải thích cho các sự kiện chúng ta quan sát được.

Ví dụ : Trời mưa thì đường sẽ ướt.

Hôm nay đường ướt.

Suy ra : Trời mưa.

- Đây là cách lập luận không đáng tin cậy, vì có rất nhiều lý do để giải thích cho một sự kiện.

Ví dụ: đường ướt có thể do xe rửa đường.

1.2.4.2. Giải quyết vấn đề (1) – Lý thuyết Gestalt

- Quá trình giải quyết vấn đề của con người bao gồm: sản xuất và tái sản xuất.
- Tái sản xuất là xây dựng từ các kinh nghiệm có trước.
- Sản xuất là dựng nên bản chất và cấu trúc hóa lại vấn đề.
- Bản chất của lý thuyết này là dịch chuyển từ các hành vi đến lý thuyết xử lý thông tin.
- Lý thuyết này không cung cấp đủ các bằng chứng để hỗ trợ lý thuyết.

1.2.4.2. Giải quyết vấn đề (2) – Lý thuyết không gian bài toán

- Không gian bài toán bao gồm các phát biểu của bài toán.
- Việc giải quyết vấn đề bao gồm việc khởi tạo các phát biểu này bằng cách dùng các phép dịch chuyển hợp lý.
- Một bài toán có một trạng thái ban đầu và một trạng thái đích. Người dùng sử dụng các phép toán để dịch chuyển từ trạng thái ban đầu đến trạng thái đích.
- Do không gian bài toán thường rất lớn, nên con người sử dụng phương pháp tìm kiếm *heuristic* để lựa chọn các phép toán thích hợp để đi đến đích.
- Kinh nghiệm sẽ cho phép con người giải quyết vấn đề một cách dễ dàng hơn khi có thể cấu trúc không gian bài toán một cách tương ứng và lựa chọn các phép toán một cách hiệu quả.

Chương 1 – GIỚI THIỆU CHUNG

1.1. Tương tác người – máy là gì

1.2. Đặc điểm tâm sinh lý con người trong quá trình giao tiếp

1.3. Phương tiện giao tiếp của máy tính

Chương 1 – GIỚI THIỆU CHUNG

1.3. Phương tiện giao tiếp của máy tính

1.3.1. Các thành phần cơ bản của máy tính

1.3.2. Thiết bị nhập văn bản và định vị trên màn hình

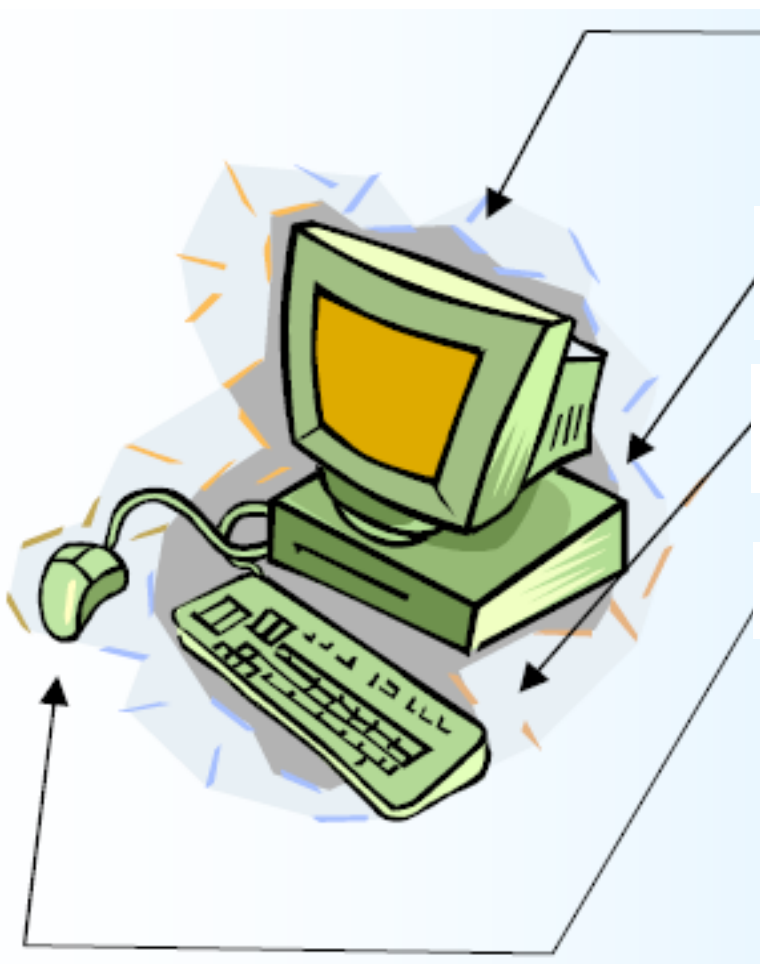
1.3.3. Thiết bị hiển thị

1.3.4. Thiết bị vào – ra đồ thị và in ấn

1.3.5. Bộ nhớ

1.3.6. Xử lý thông tin

1.3.1. Các thành phần cơ bản của máy tính (1)



- Màn hình - Định nghĩa(Thimbleby):
“máy tính là một thành phần tham gia tương tác, thực hiện một chương trình”.
- CPU
- Bàn phím - Các thành phần cơ bản của máy tính mà HCI quan tâm:
- Chuột
- + Thiết bị vào/ra
 - + Bộ nhớ
 - + Bộ xử lý

1.3.1. Các thành phần cơ bản của máy tính (2)

* Hai cách xử lý thông tin trong máy tính:

➤ Xử lý thông tin theo lô (batch):

- Thông tin theo lô có khuôn dạng nhất định.
- Khối lượng dữ liệu lớn, yêu cầu nhập dữ liệu càng nhanh càng tốt.
- Thường là giao tiếp giữa máy với máy.

➤ Xử lý thông tin trực tuyến (on – line):

- Người dùng ngồi trước máy tính, thông tin được đưa vào theo lời nhắc.
- Máy tính là công cụ quản lý và nhận thông tin.
- Thông tin không được bổ sung vào kho lưu trữ của máy tính.
- HCI quan tâm đến cách xử lý thông tin này.

1.3.2. Thiết bị nhập văn bản và định vị trên màn hình (1) – Bàn phím

➤ Bàn phím QWERTY:

- Thông dụng nhất.
- Thiết kế dựa trên kiểu bàn phím máy chữ.
- Gồm 4 nhóm: nhóm phím chữ cái, nhóm phím số, nhóm phím di chuyển con trỏ, nhóm phím đặc biệt (phím chức năng, phím điều khiển).



1.3.2. Thiết bị nhập văn bản và định vị trên màn hình (2) – Bàn phím

➤ Bàn phím chữ cái:

- Các phím được bố trí theo thứ tự chữ cái.
- Thường sử dụng cho các thiết bị điện tử cá nhân bỏ túi.

1.3.2. Thiết bị nhập văn bản và định vị trên màn hình (3) – Bàn phím

- Bàn phím DVORAK (đặt tên theo tên người phát minh):
 - Thiết kế tương tự kiểu QWERTY, các chữ cái được gán cho các phím khác nhau.
 - Giúp người dùng đạt tốc độ gõ phím cao, giảm sự mệt mỏi cho người dùng.



1.3.2. Thiết bị nhập văn bản và định vị trên màn hình (4) – Bàn phím

➤ Bàn phím CHORD:

- Chỉ có 4 hoặc 5 phím, chữ cái được tạo ra khi nhấn đồng thời một số phím.
- Ưu điểm: kích thước nhỏ, học nhanh.
- Nhược điểm: thiếu tính tự nhiên của cách bố trí phím và số lượng phím.
- Có thể sử dụng cho các thiết bị điện tử bỏ túi.

1.3.2. Thiết bị nhập văn bản và định vị trên màn hình (5) – Bộ nhận dạng

➤ Bộ nhận dạng chữ viết:

- Nhận dạng chữ viết tay và biến đổi sang văn bản.
- Độ chính xác không cao, nhưng đang ngày càng được cải tiến.
- Được sử dụng trong các thiết bị bỏ túi, như: PDA, ...

➤ Bộ nhận dạng tiếng nói:

- Là một dạng của nhận dạng văn bản.
- Thiết bị phức tạp.
- Giá thành cao.
- Ưu điểm: tự nhiên, tỉ lệ nhận dạng cao (90%).

1.3.2. Thiết bị nhập văn bản và định vị trên màn hình (6) – Chuột

- Là thiết bị trỏ và định vị.
- Thường được thiết kế với kích thước nhỏ, có từ 1 đến 3 phím.
- Một số loại chuột: chuột bi, chuột quang, ...



1.3.2. Thiết bị nhập văn bản và định vị trên màn hình (7) – Thiết bị định vị hai chiều

➤ *Tracball:*

- Cơ chế phát hiện chuyển động: giống con chuột.
- Không cần không gian khi thao tác.
- Độ chính xác cao hơn chuột.
- Khó khi dịch chuyển dài.



1.3.2. Thiết bị nhập văn bản và định vị trên màn hình (8) – Thiết bị định vị hai chiều

➤ Joystick:

- Gồm một hộp mỏng với một cần.
- Sự chuyển động của cần tạo nên sự chuyển động tương ứng của con trỏ.
- Thường dùng trong các thiết bị trò chơi điện tử.



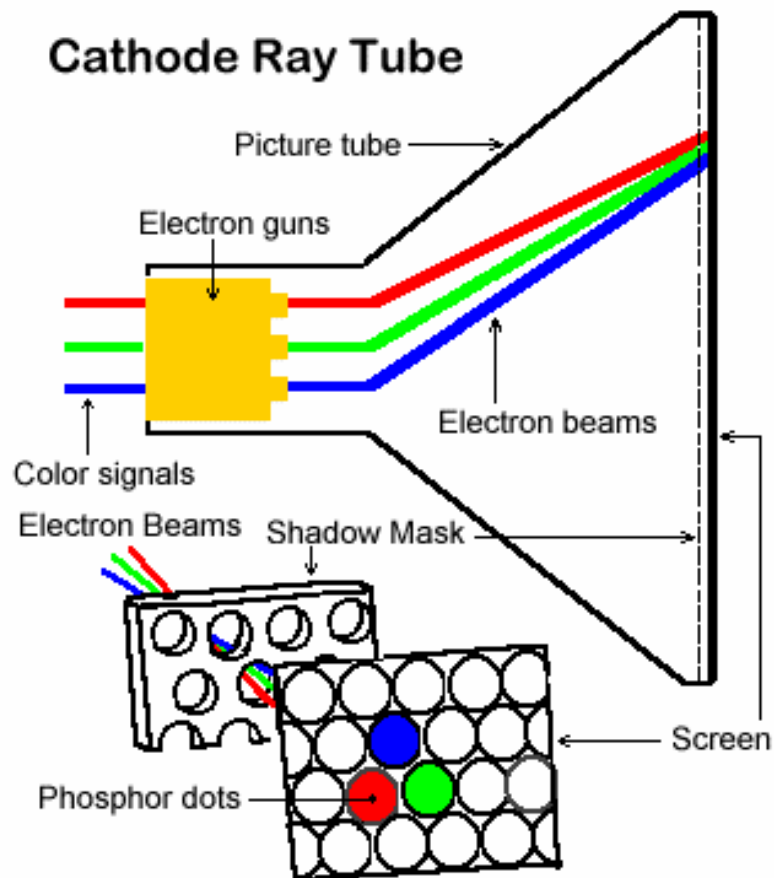
1.3.3. Thiết bị hiển thị (1) – Màn hình CRT

➤ Ưu điểm:

- Thời gian đáp ứng nhanh.
- Giá thành thấp.

➤ Nhược điểm:

- Kích thước lớn, cồng kềnh.
- Hình ảnh bị nhấp nháy, gây mỏi mắt.



1.3.3. Thiết bị hiển thị (2) – Màn hình LCD

- Kích thước nhỏ gọn, phù hợp với các máy tính xách tay hay thiết bị cá nhân.
- Tiêu tốn ít năng lượng hơn loại CRT.
- Không có phóng xạ.
- Ít nhấp nháy hơn loại CRT.
- Giá thành cao hơn.



1.3.4. Thiết bị vào – ra đồ thị và in ấn (1) – Máy in

- Là thiết bị hỗ trợ máy tính tạo ra văn bản trên giấy.
- Một ảnh hay một ký tự được tạo ra trên giấy bằng một chuỗi các chấm đen.
- Các loại máy in:
 - Máy in ma trận điểm: độ phân giải khoảng 300 dpi (dot per inch).
 - Máy in phun: độ phân giải tương tự máy in ma trận điểm.
 - Máy in lade: độ phân giải cao, từ 600 đến 1200 dpi.



1.3.4. Thiết bị vào – ra đồ thị và in ấn (2) – Máy quét, máy chiếu

➤ *Máy quét:*

- Nhận văn bản trên giấy và chuyển thành văn bản điện tử.
- Sử dụng thiết bị nhận dạng ký tự quang học để chuyển ảnh sang dạng bitmap.



➤ *Máy chiếu:*

- Nhận văn bản điện tử và chuyển thành hình ảnh.



1.3.5. Bộ nhớ

- Gồm hai loại: STM và LTM.
- Bộ nhớ ngắn hạn STM: (RAM)
 - Tốc độ truy cập nhanh, khoảng 100ns.
 - Tốc độ truyền lớn, khoảng 10 MB/s.
 - Dung lượng nhỏ, từ 32 MB tới 256 MB.
 - Phần lớn là loại bay hơi. Một số RAM không bay hơi có giá thành cao.
- Bộ nhớ dài hạn LTM:
 - Là bộ nhớ ngoài: đĩa từ, đĩa CD,...
 - Dung lượng thường lớn.
 - Tốc độ truy cập thường chậm hơn so với STM.

1.3.6. Xử lý thông tin (1)

- Tốc độ xử lý ảnh hưởng quan trọng đến hệ thống tương tác.
- Tốc độ xử lý quá chậm: thông tin phản hồi từ hệ thống đến người dùng chậm, ảnh hưởng đến quá trình giao tiếp.

Ví dụ: người dùng nhấn chuột trên một đối tượng mà chưa thấy phản ứng, tiếp tục nhấn nhiều lần, có thể gây treo máy.

- Tốc độ xử lý quá nhanh: người dùng không theo kịp và không biết hệ thống đã đáp ứng những gì.

Ví dụ: sử dụng một máy tính tốc độ thấp để viết chương trình trình diễn một quá trình làm việc. Khi chạy chương trình trên một máy tính tốc độ nhanh quá, có thể không quan sát được toàn bộ quá trình.

1.3.6. Xử lý thông tin (2)

- Một số yếu tố ảnh hưởng đến hiệu suất của các hệ thống tương tác:
- Tốc độ tính toán.
 - Tốc độ lưu trữ, tốc độ lấy dữ liệu từ các thiết bị lưu trữ.
 - Tốc độ xử lý đồ họa.
 - Tốc độ truyền tin trên mạng.

Chương 2 – TỔ CHỨC TƯƠNG TÁC

2.1. Mô hình tương tác

2.2. Các dạng tương tác

2.3. Giao diện WIMP

2.4. Điều khiển trực tiếp

2.5. Thiết kế và bố trí màn ảnh

Chương 2 – TỔ CHỨC TƯƠNG TÁC

2.1. Mô hình tương tác

2.1.1. *Giai đoạn tương tác*

2.1.2. *Chu trình thực hiện – đánh giá*

2.1.3. *Framework*

2.1.4. *Framework và HCI*

2.1.5. *Tương quan người – môi trường*

2.1.1. Giai đoạn tương tác (1)

- Tương tác: là sự giao tiếp giữa người dùng và hệ thống.
- Máy tính: là công cụ thực hiện, hay công cụ hỗ trợ nhiệm vụ.
- HCI quan tâm: dạng tương tác điều khiển trực tiếp. Người dùng cung cấp các chỉ thị và nhận hồi đáp từ hệ thống.
- Các mô hình tương tác: tìm ra và đánh giá các thành phần của sự tương tác.
- Các dạng tương tác: đánh giá xem các dạng tương tác này có hiệu quả hay không.

2.1.1. Giai đoạn tương tác (2) – Các mô hình tương tác

- Giao diện: làm nhiệm vụ truyền tải một cách hiệu quả giữa con người và máy tính, đảm bảo tương tác thành công.
- Sự truyền tải này có thể bị trục trặc ở một số điểm vì nhiều lý do.
- Sử dụng mô hình tương tác giúp:
 - Hiểu chính xác điều gì sẽ diễn ra trên tương tác.
 - Hiểu nguồn gốc những khó khăn khi tương tác.
- Hai mô hình tương tác:
 - Mô hình “chu trình thực hiện – đánh giá” (Normal)
 - Mô hình Framework - khung tương tác (Abowd và Beale)

2.1.2. Chu trình thực hiện – đánh giá (1)

- Mô hình này có ảnh hưởng nhiều nhất trong HCI vì nó rất gần với hiểu biết cụ thể của chúng ta về tương tác giữa người dùng và máy tính.
- Hoạt động của người sử dụng trong chu trình:
 - Xây dựng một kế hoạch hành động.
 - Thực hiện kế hoạch đó thông qua giao diện máy tính.
 - Quan sát giao diện máy tính khi kế hoạch hoặc một phần kế hoạch được thực hiện, để:
 - + Đánh giá kết quả thực hiện được của kế hoạch.
 - + Xác định các hành động tiếp theo.

2.1.2. Chu trình thực hiện – đánh giá (2) – Các giai đoạn

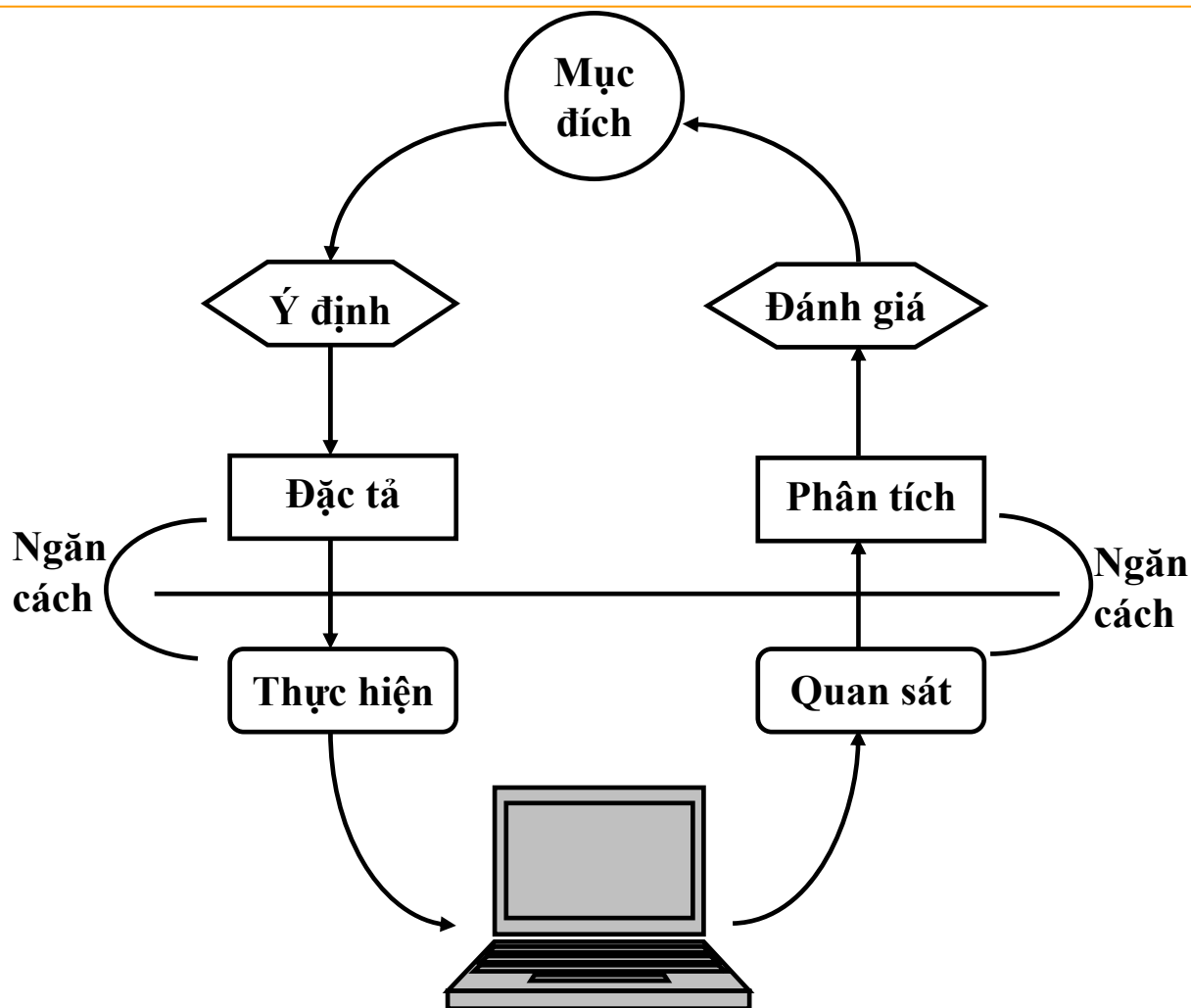
* *Giai đoạn thực hiện:*

- Thiết lập mục đích
- Sắp xếp ý định
- Đặc tả dãy hành động
- Thực hiện hành động

* *Giai đoạn đánh giá:*

- Quan sát trạng thái hệ thống
- Phân tích trạng thái hệ thống
- Đánh giá trạng thái hệ thống dựa trên các mục đích và các ý định.

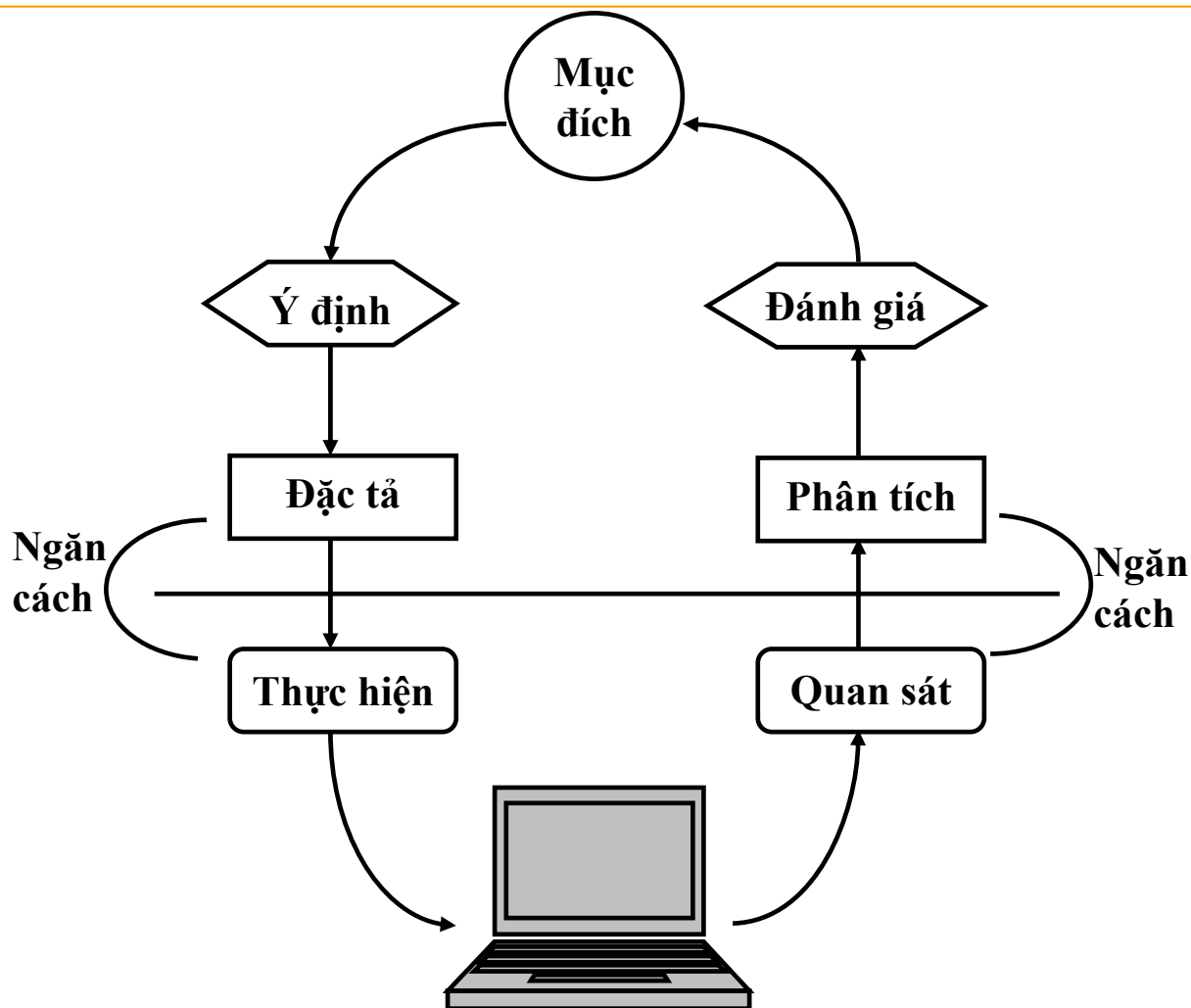
2.1.2. Chu trình thực hiện – đánh giá (3) – Các giai đoạn



* Các giai đoạn thực hiện:

- Truyền tải mục đích ban đầu thành một ý định để thực hiện một việc gì đó.
- Chuyển ý định thành một dãy các hành động (nhằm thỏa mãn ý định).
- Thực hiện dãy hành động.

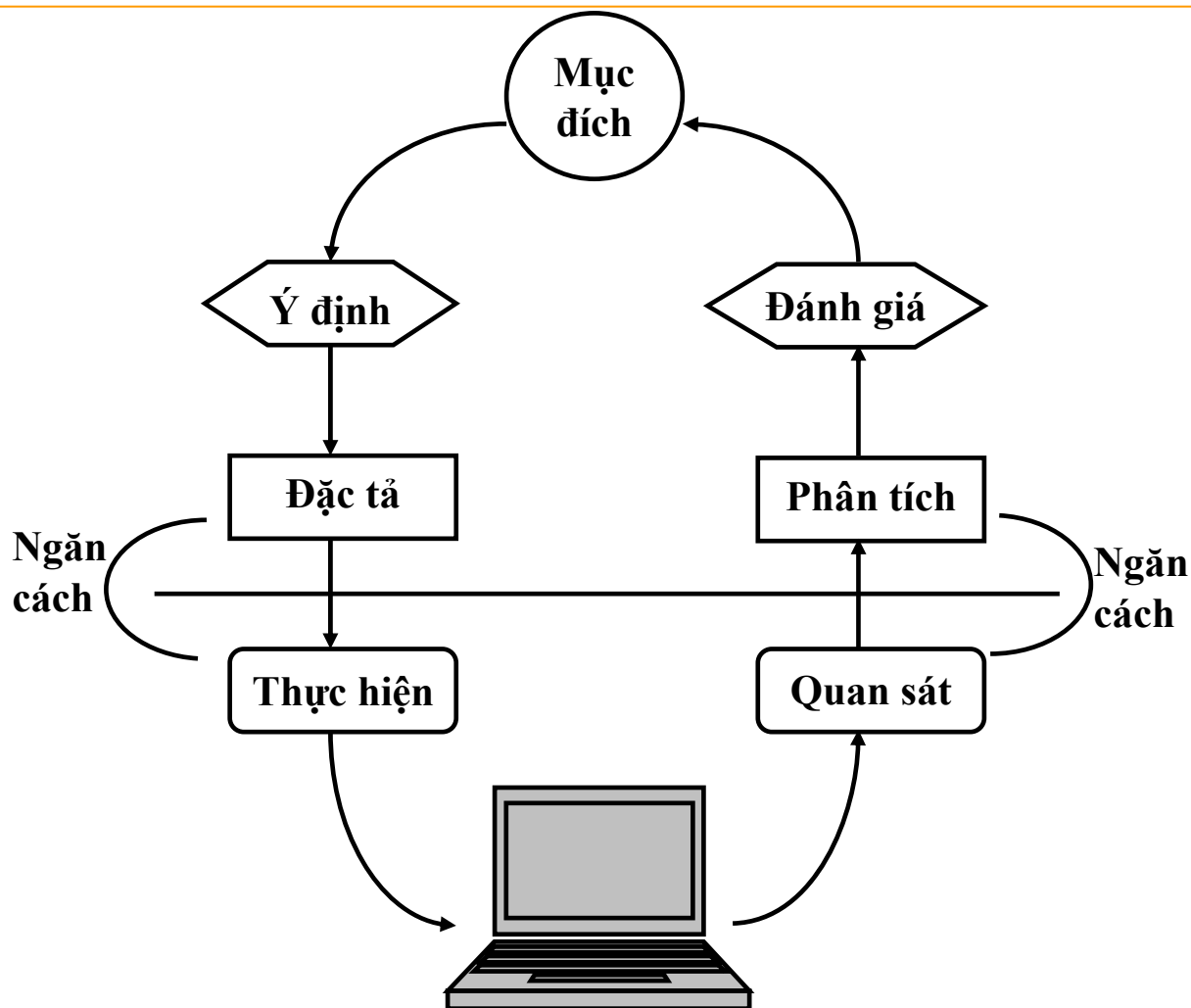
2.1.2. Chu trình thực hiện – đánh giá (4) – Các giai đoạn



* Các giai đoạn đánh giá:

- Quan sát trạng thái mới của hệ thống.
- Phân tích trạng thái hệ thống theo kỳ vọng của mình.
- Nếu trạng thái hệ thống phản ánh đúng mục đích: tương tác thành công.
- Ngược lại, người dùng xây dựng mục đích mới, lặp lại chu trình.

2.1.2. Chu trình thực hiện – đánh giá (5) – Trở ngại khi tương tác



- *Ngăn cách trong thực hiện*: là sự khác biệt giữa những hành động mà người dùng dự định làm và các hành động mà hệ thống chấp nhận được.

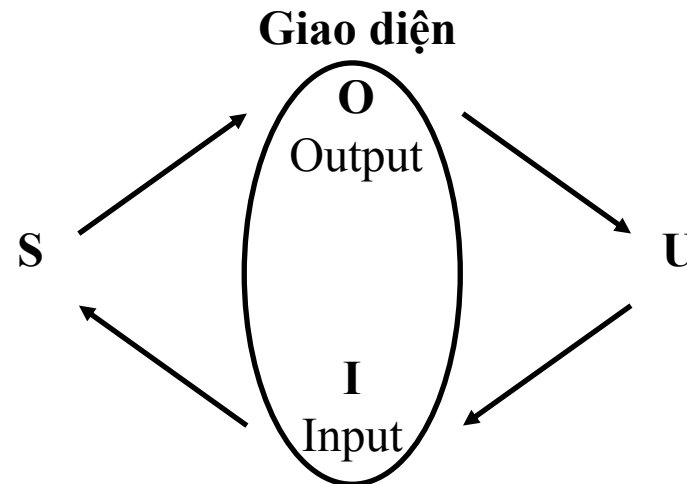
- *Ngăn cách trong đánh giá*: là sự khác biệt giữa những thứ mà hệ thống trình diễn với những điều mà người dùng mong đợi.

2.1.2. Chu trình thực hiện – đánh giá (6)

- Mô hình này rất rõ ràng và trực quan, là một công cụ hữu ích trong quá trình tìm hiểu sự tương tác.
- Một số nhược điểm của mô hình chu trình thực hiện – đánh giá:
 - Chỉ quan tâm đến hệ thống như một giao diện.
 - Chỉ tập trung đến quan điểm của người dùng về sự tương tác.
 - Không quan tâm đến sự đối thoại của hệ thống thông qua giao diện.

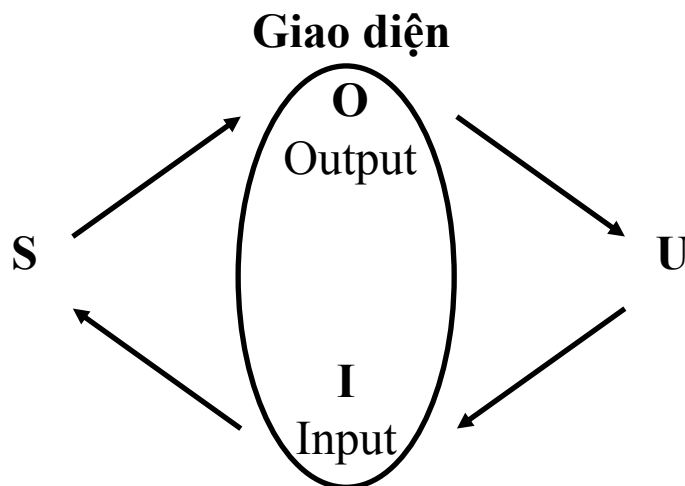
2.1.3. Framework (1)

- Mô hình này được phát triển dựa trên ý tưởng của Norman về mô hình chu trình thực hiện – đánh giá.
- Hệ thống được mô tả một cách tường minh, hiện thực hơn.
- Các thành phần của hệ thống:
 - Hệ thống (S)
 - Người dùng (U)
 - Đầu vào (I)
 - Đầu ra (O)



2.1.3. Framework (2)

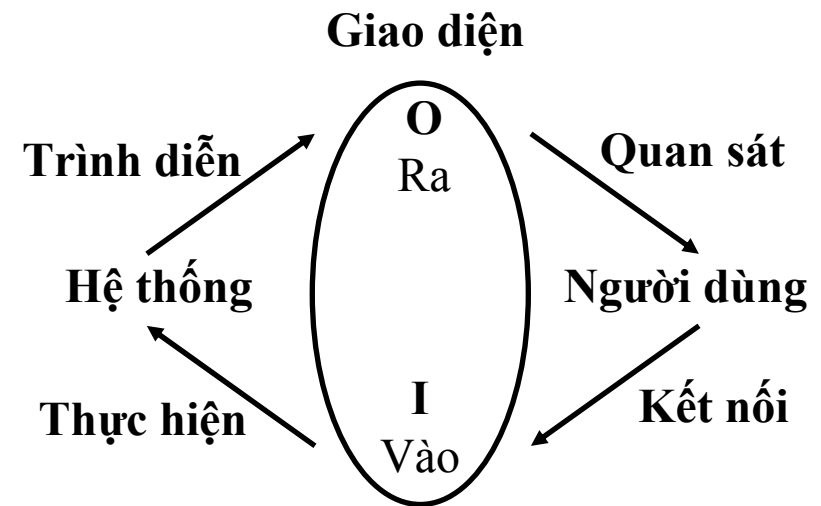
- Mỗi thành phần dùng ngôn ngữ riêng.
- Các thành phần vào, ra cũng có các ngôn ngữ riêng.
- Đầu vào và đầu ra kết hợp tạo thành giao diện.



2.1.3. Framework (3) – Các bước trong chu trình

- Gồm 4 bước, mỗi bước tương ứng với một dịch chuyển từ thành phần này đến thành phần kia:

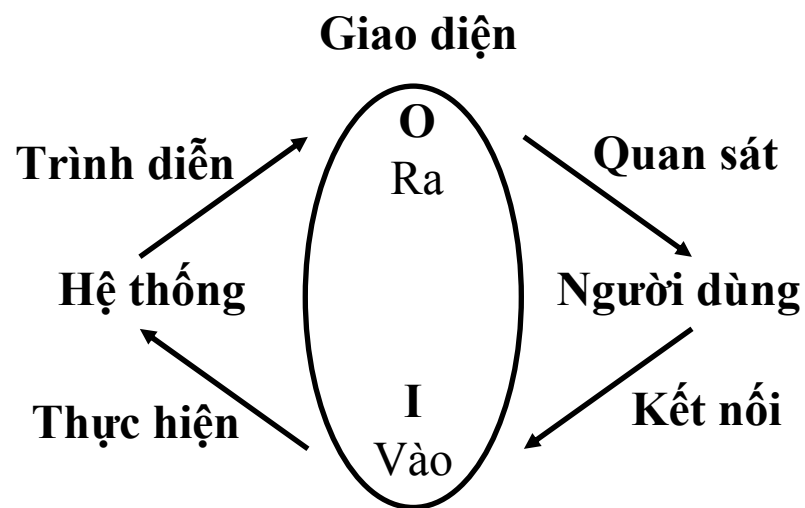
- Kết nối
- Thực hiện
- Trình diễn
- Quan sát



2.1.3. Framework (4) – Các bước trong chu trình

- *Kết nối:*

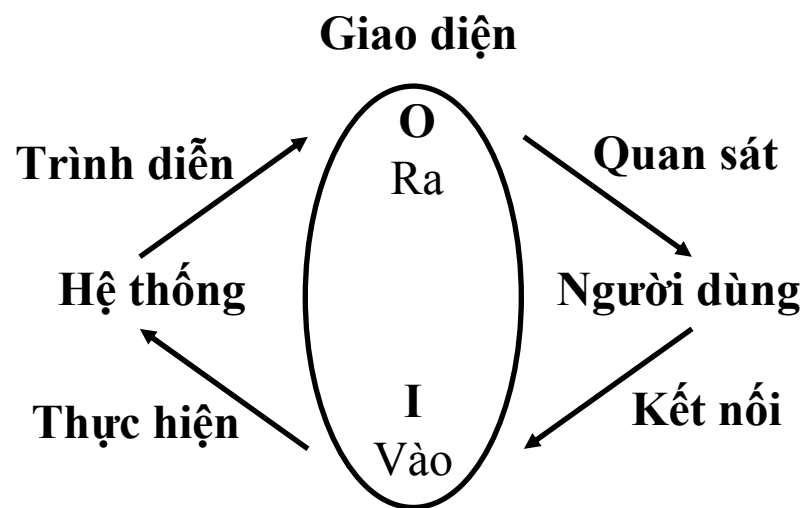
- Là bước đầu tiên.
- Người sử dụng thiết lập một mục đích và một nhiệm vụ để đạt được mục đích đó.
- Người sử dụng giao tiếp với máy thông qua đầu vào.
- Nhiệm vụ phải được dịch chuyển thành các kích thích đầu vào.



2.1.3. Framework (5) – Các bước trong chu trình

- *Thực hiện:*

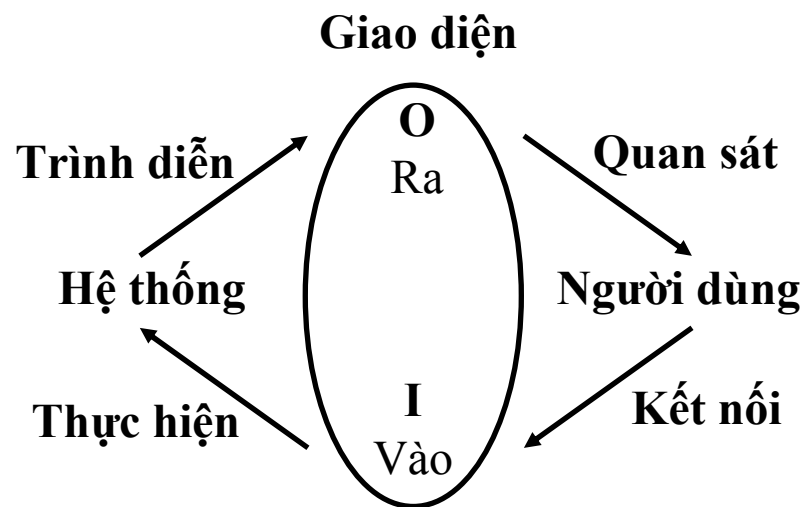
- Các đáp ứng của đầu vào được dịch chuyển để kích thích hệ thống.
- Ngôn ngữ đầu ra được dịch thành ngôn ngữ của hệ thống để hệ thống có thể hiểu.
- Hệ thống tự chuyển đổi theo các thao tác được dịch từ đầu vào.
Kết thúc giai đoạn thực hiện.



2.1.3. Framework (6) – Các bước trong chu trình

- *Trình diễn:*

- Khi thực hiện nhiệm vụ, hệ thống sẽ chuyển sang trạng thái mới.
- Trạng thái mới này cần được thông báo cho người dùng.
- Đáp ứng của hệ thống được dịch chuyển thành các kích thích ở đầu ra (khái niệm hoặc đặc điểm của đầu ra).

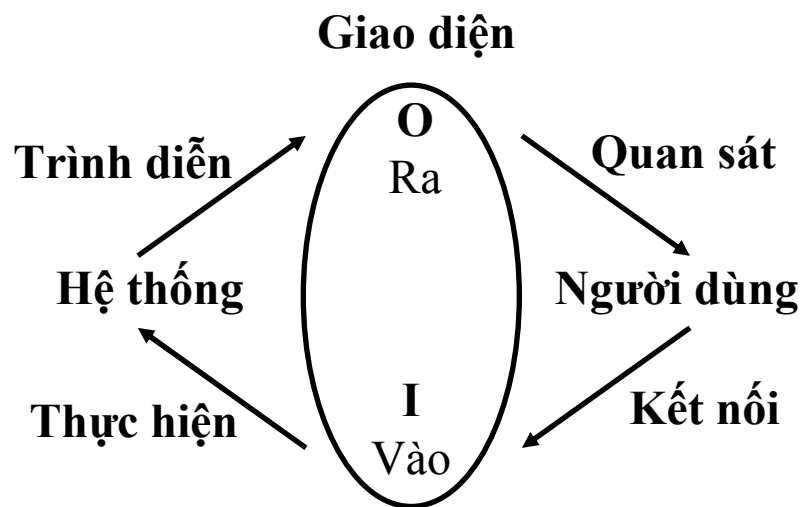


2.1.3. Framework (7) – Các bước trong chu trình

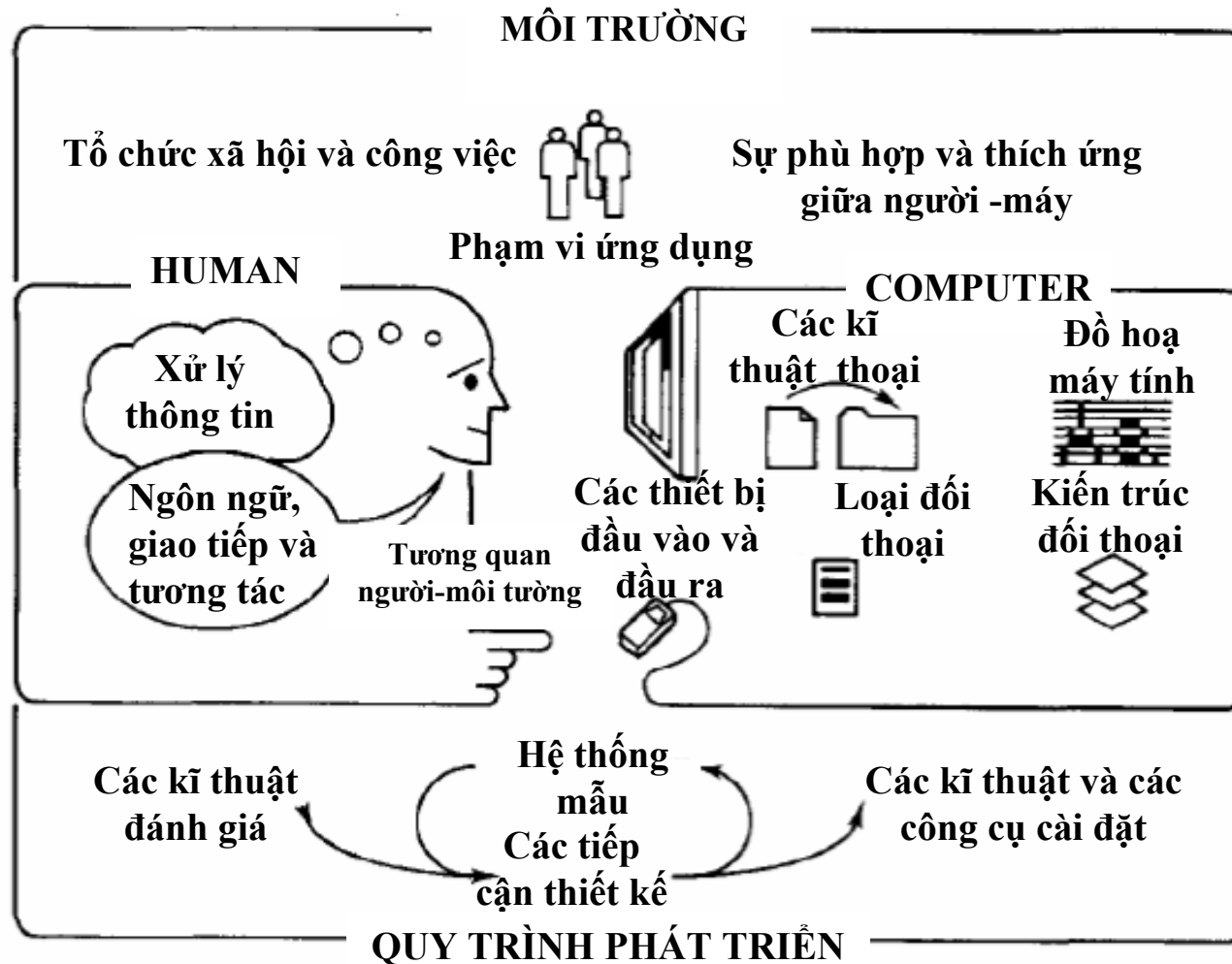
- *Quan sát:*

- Người sử dụng quan sát đầu ra.
- Dựa trên kết quả quan sát, đánh giá các kết quả của tương tác so với mục đích ban đầu.

Kết thúc giai đoạn đánh giá và kết thúc chu trình tương tác.



2.1.4. Khung tương tác người – máy (1)



2.1.4. Khung tương tác người – máy (2)

- *Môi trường và ngữ cảnh:*
 - Là tập những yếu tố mà máy tính cung cấp.
 - Bao gồm:
 - + Phạm vi ứng dụng: cung cấp các chức năng và giao tiếp cho người dùng để hoàn thành nhiệm vụ.
 - + Các tổ chức xã hội và công việc: tương tác với các ứng dụng.
 - + Sự phù hợp và thích ứng giữa người và máy.

2.1.4. Khung tương tác người – máy (3)

- *Con người:*
 - Là trung tâm của việc thiết kế phần mềm và tương tác.
 - Con người được hiểu là một bộ xử lý thông tin.
 - Thông qua ngôn ngữ, giao tiếp và tương tác, con người nhận thức và hiểu cách thức hành động để đạt mục đích.
 - Tương quan người – môi trường:
 - + Liên quan đến môi trường và điều kiện làm việc.
 - + Liên quan đến khả năng nhận thức và giới hạn vật lý của con người.

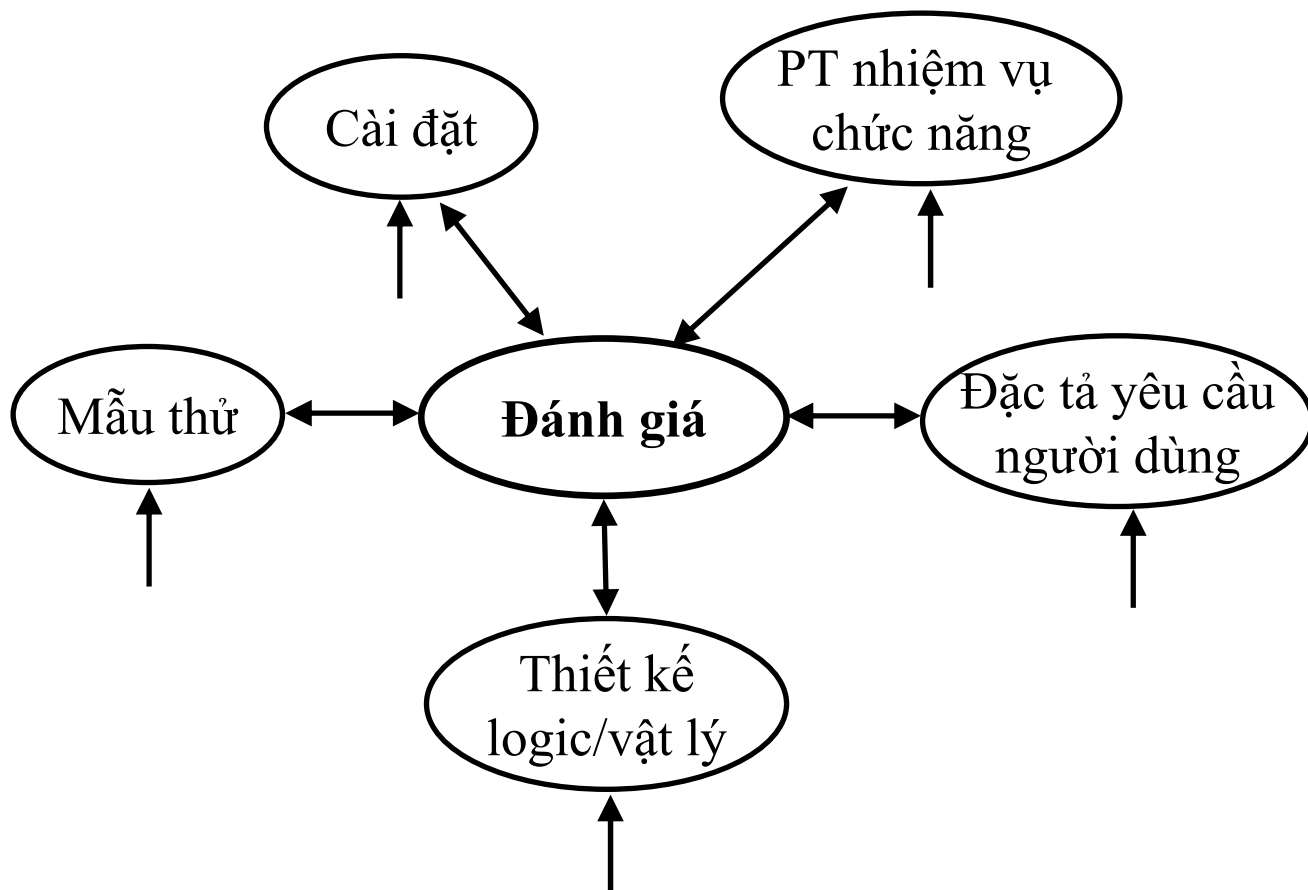
2.1.4. Khung tương tác người – máy (4)

- *Máy tính*: quan tâm đến các vấn đề sau:
 - Các thiết bị đầu vào và đầu ra: nhập và hiển thị thông tin.
 - Các kỹ thuật đối thoại: thực hiện tương tác.
 - Các loại đối thoại.
 - Kiến trúc đối thoại: mô tả đối thoại.
 - Đồ họa máy tính: biểu diễn đối thoại.

2.1.4. Khung tương tác người – máy (5)

- Quy trình phát triển:

- Thiết kế và phát triển tương tác người – máy diễn ra qua nhiều giai đoạn, gọi là vòng đời.
- Vòng đời hình sao của HCI:



2.1.5. Tương quan người – môi trường (1)

- *Công thái học:*
 - Là một bộ phận của tâm lý học ứng dụng.
 - Nghiên cứu các đặc điểm tự nhiên của tương tác: cách thiết kế các điều khiển, môi trường vật lý mà tương tác xảy ra, và chất lượng hiển thị và chất lượng vật lý của màn hình.
- *Mục đích của công thái học:* cải tiến thiết kế thái học để có thể áp dụng cho máy tính, cả về phần cứng và phần mềm.
- *Các vấn đề về công thái học liên quan đến HCI:*
 - Sắp xếp các điều khiển và hiển thị.
 - Môi trường vật lý của tương tác.
 - Các vấn đề về sức khỏe.
 - Sử dụng màu sắc.

2.1.5.1. Sắp xếp các điều khiển và hiển thị (1)

- Các nút điều khiển và các phần trên màn hình hiển thị nên được sắp xếp một cách logic để người sử dụng dễ dàng truy cập.
- Thiết kế hiển thị vật lý phù hợp là vấn đề rất quan trọng với các ứng dụng đòi hỏi sự an toàn, chính xác cao: điều khiển sản xuất, điều khiển hàng không, điều khiển không lưu,...
- Giao diện hệ thống phải được sắp xếp phù hợp với vị trí của người sử dụng. Ví dụ: người sử dụng phải có khả năng truy cập đến tất cả các điều khiển và quan sát được tất cả các hiển thị mà không phải di chuyển vị trí.
- Nên thiết kế những phần hiển thị quan trọng sao cho dễ nhìn thấy nhất.

2.1.5.1. Sắp xếp các điều khiển và hiển thị (2) – Các cách tổ chức

- Tổ chức theo chức năng: các điều khiển và hiển thị có chức năng tương tự nhau thường được đặt cạnh nhau.
- Tổ chức theo kiểu tuần tự: tổ chức các điều khiển và hiển thị sao cho có thể phản ánh thứ tự sử dụng của chúng trong một tương tác điển hình.
- Tổ chức theo tần suất xuất hiện: điều khiển nào được sử dụng nhiều nhất sẽ là điều khiển có thể truy cập dễ dàng nhất.

2.1.5.2. Môi trường vật lý của tương tác

- Môi trường vật lý có thể ảnh hưởng đến việc tiếp nhận, sức khỏe và an toàn của người sử dụng.
- Khối lượng người sử dụng:
 - Khi số lượng người sử dụng ít nhất: mọi người đều có khả năng truy cập đến tất cả các điều khiển.
 - Khi số lượng người sử dụng lớn nhất: mọi người không bị gò bó trong môi trường chật hẹp.
- Phải thiết kế sao cho mọi người sử dụng đều có thể nhìn thấy tất cả phần hiển thị một cách thoải mái nhất.

2.1.5.3. Các vấn đề về sức khỏe (1)

- Thiết kế giao diện phải quan tâm đến sức khỏe và sự an toàn của người sử dụng.
- Một số nhân tố trong môi trường vật lý không chỉ ảnh hưởng đến sức khỏe và sự an toàn của người sử dụng mà còn trực tiếp ảnh hưởng đến chất lượng tương tác và hiệu quả làm việc của người sử dụng.
- Các nhân tố đó bao gồm: vị trí vật lý, nhiệt độ, ánh sáng, tiếng ồn, thời gian.

2.1.5.3. Các vấn đề về sức khỏe (2)

- Vị trí vật lý: thiết kế nên đảm bảo cho mọi người sử dụng đều có khả năng truy cập đến tất cả các điều khiển và đều xem được toàn bộ mọi hiển thị.
- Nhiệt độ: nhiệt độ làm việc phải phù hợp với cơ thể, khi nhiệt độ quá cao hoặc quá thấp, hiệu quả làm việc sẽ giảm và người sử dụng sẽ không thể tập trung vào công việc được.
- Ánh sáng: nên đảm bảo mức ánh sáng phù hợp để người sử dụng không phải điều tiết mắt để nhìn màn hình. Đặt nguồn ánh sáng ở vị trí thích hợp để không bị chói.

2.1.5.3. Các vấn đề về sức khỏe (3)

- Tiếng ồn: mức tiếng ồn nên duy trì ở ngưỡng vừa phải. Tiếng ồn vượt mức cho phép có ảnh hưởng xấu đến sức khỏe người sử dụng, gây đau tai, thậm chí mất khả năng nghe.
- Thời gian: cần kiểm soát thời gian người sử dụng làm việc với hệ thống. Thời gian sử dụng quá dài có thể ảnh hưởng xấu đến sức khỏe người sử dụng.

2.1.5.4. Sử dụng màu sắc (1)

- Một số giới hạn liên quan đến màu sắc của hệ thống thị giác: số lượng màu sắc mà mắt có thể phân biệt được; khó khi thu nhận màu xanh da trời,...
- Các màu được sử dụng trên màn hình nên rõ ràng.
- Không nên dùng màu xanh da trời để hiển thị các thông tin quan trọng.
- Không nên sử dụng màu như một chỉ dẫn duy nhất, nên có các thông tin chỉ dẫn phụ.
- Sử dụng màu nên theo quy ước chung. Ví dụ: màu đỏ - cảnh báo; màu xanh lá cây: hoạt động bình thường; màu vàng: hỗ trợ hoặc dự phòng.

2.1.5.4. Sử dụng màu sắc (2)- Một số kết hợp màu

Màu nền sau	Các màu nền trước nên dùng	Các màu nền trước nên tránh
Trắng	Đen, xanh da trời đậm, đỏ	Xám nhạt, màu lục lam
Xanh da trời	trắng, vàng, màu lục lam	Xanh lá cây
Hồng	trắng, vàng	Xanh lá cây, lục lam
đỏ	vàng, trắng	lục lam, xanh lá cây
Vàng	đỏ, xanh da trời, đen	lục lam
Xanh lá cây	Đen, đỏ, xanh da trời	lục lam, hồng, vàng
Lục lam	Xanh da trời, đen, đỏ	Xanh lá cây, vàng, trắng
Xám nhạt	Đen, xanh da trời đậm, hồng đậm	Xanh lá cây, lục lam, vàng
Xám	Vàng, trắng, xanh da trời	Xám đậm, lục đậm
Xám đậm	lục lam, vàng, xanh lá cây	đỏ, xám
Đen	trắng, lục, xanh lá cây, vàng	lục đậm
Xanh da trời đậm	Vàng, trắng, hồng, xanh lá cây	Xanh lá cây đậm
Hồng đậm	Xanh lá cây, vàng, trắng	lục đậm

Chương 2 – TỔ CHỨC TƯƠNG TÁC

2.1. Mô hình tương tác

2.2. Các dạng tương tác

2.3. Giao diện WIMP

2.4. Điều khiển trực tiếp

2.5. Thiết kế và bố trí màn ảnh

Chương 2 – TỔ CHỨC TƯƠNG TÁC

2.2. Các dạng tương tác

2.2.1. Sử dụng dòng lệnh

2.2.2. Sử dụng thực đơn (menu)

2.2.3. Sử dụng ngôn ngữ tự nhiên

2.2.4. Đối thoại

2.2.5. Sử dụng form và bảng tính

2.2.6. Chọn và nhấn

2.2.1. Sử dụng dòng lệnh (1)

- Giao diện dòng lệnh là kiểu giao tiếp đối thoại tương tác đầu tiên được sử dụng trong các hệ thống máy tính.
- Hiện loại giao diện này vẫn được sử dụng rộng rãi.

Ví dụ: MS DOS, UNIX

- Giao diện này cung cấp phương tiện để biểu diễn trực tiếp các chỉ thị đến máy tính.
- Người dùng nhập lệnh để thực hiện yêu cầu của mình bằng một số phương tiện: sử dụng các phím chức năng, các phím tắt, các ký tự riêng lẻ, các câu lệnh rút gọn hoặc đầy đủ.

2.2.1. Sử dụng dòng lệnh (2)

```
-- Logical Library Mappings
WORK > TB_LIB
TB_LIB   : /prj/libs/tb_lib
DUT_LIB  : /prj/libs/dut_lib
vendor   : ../../vendor/libs

-- Simulator Variable Settings
ASSERT_STOP = ERROR
ASSERT_IGNORE = WARNING
TIME_RESOLUTION = 10 ps
```


2.2.1. Sử dụng dòng lệnh (3)

- *Ưu điểm:*
 - Rất nhanh, do truy cập trực tiếp đến các chức năng của hệ thống.
 - Giao tiếp linh hoạt, vì các lệnh thường có các tùy chọn, người dùng có thể sử dụng trong các tình huống khác nhau.
- *Nhược điểm:*
 - Khó học, khó sử dụng, do người dùng phải nhớ lệnh và cú pháp lệnh.
 - Hệ thống thường hay bị lỗi khi giao tiếp.

2.2.2. Sử dụng thực đơn (menu) (1)

- Giao diện menu cung cấp một tập các lựa chọn có thể cho người dùng, tập lựa chọn này được hiện trên màn hình.

PHƯƠNG TIỆN THANH TOÁN

Lựa chọn một trong các phương tiện thanh toán sau:

1. Tiền mặt
2. Séc
3. Thẻ tín dụng
4. Hóa đơn

2.2.2. Sử dụng thực đơn (menu) (2)

- Người dùng lựa chọn một mục (tương ứng một công việc) bằng cách sử dụng:
 - Các phím con trỏ
 - Phím tắt
 - Nhấn vào một ký tự (hay chữ số)
 - Sử dụng chuột

PHƯƠNG TIỆN THANH TOÁN

Lựa chọn một trong các phương tiện thanh toán sau:

1. Tiền mặt
2. Séc
3. Thẻ tín dụng
4. Hóa đơn

2.2.2. Sử dụng thực đơn (menu) (3)

- *Ưu điểm:*
 - Người dùng được gợi ý và không phải nhớ, vì các lựa chọn được quan sát trực tiếp trên màn hình.
 - Ít lỗi hơn so với giao diện dòng lệnh.
 - Phù hợp với người dùng không thường xuyên, tốn ít công đào tạo.
 - Kiểu giao diện menu có thể sử dụng trong môi trường văn bản và cả môi trường đồ họa.

2.2.2. Sử dụng thực đơn (menu) (4)

- *Nhược điểm:*
 - Chiếm không gian màn hình.
 - Gây khó khăn cho người sử dụng khi có quá nhiều lựa chọn với nhiều cấp độ khác nhau.
 - Thiếu tính mềm dẻo, không thể cung cấp gì thêm ngoài các lựa chọn được liệt kê.
 - Tốc độ chậm đối với người dùng thành thạo.
 - Không cung cấp đầy đủ các tính năng tương tác.

2.2.3. Sử dụng ngôn ngữ tự nhiên (1)

- Giao tiếp bằng ngôn ngữ tự nhiên là một dạng giao tiếp hấp dẫn nhất giữa người dùng và máy tính.
- Ngôn ngữ tự nhiên bao gồm: tiếng nói, chữ viết.
- Người dùng sử dụng ngôn ngữ hàng ngày để nhập lệnh thay vì phải nhớ câu lệnh hay thứ tự menu.

2.2.3. Sử dụng ngôn ngữ tự nhiên (2)

- Khó khăn:
 - + Sự nhập nhằng của ngôn ngữ tự nhiên (về cú pháp, cấu trúc,...) làm cho một câu có thể không rõ ràng, gây nên sự khó hiểu cho máy.
 - + Sự không rõ ràng trong nghĩa của từ cũng gây khó hiểu cho máy. Con người dựa vào ngữ cảnh để phân loại nhập nhằng này. Máy tính thì không thể.
- Giao tiếp bằng ngôn ngữ tự nhiên có thể sử dụng trong một số lĩnh vực, với tập từ ngữ nhỏ.
- Hiện đang được quan tâm nghiên cứu.

2.2.4. Đối thoại (1) – Đối thoại hỏi đáp

- Đối thoại hỏi đáp là một cơ chế đơn giản, cung cấp đầu vào cho một ứng dụng trong một lĩnh vực cụ thể nào đó.
- Người dùng phải trả lời một loạt các câu hỏi của máy tính, sau mỗi câu trả lời chính xác, quá trình tương tác từng bước được thực hiện.
- Một số dạng câu hỏi:
 - Dạng Yes/No
 - Dạng đa lựa chọn
 - Dạng nhấn số

2.2.4. Đối thoại (2) – Đối thoại hỏi đáp

- *Ưu điểm:*
 - Tự nhiên.
 - Dễ học, dễ dùng, dễ thiết kế.
- *Nhược điểm:* chức năng và khả năng hạn chế.
- Rất phù hợp cho người dùng mới và ít kinh nghiệm.
- Sử dụng nhiều trong các hệ thống thông tin, các chương trình cài đặt.

2.2.4. Đối thoại (3) – Đối thoại truy vấn

- Ngôn ngữ truy vấn được sử dụng để tạo ra các truy vấn nhằm khôi phục thông tin từ một cơ sở dữ liệu.
- Sử dụng các câu kiểu ngôn ngữ tự nhiên, nhưng yêu cầu có cú pháp đặc biệt, và yêu cầu những hiểu biết về cấu trúc cơ sở dữ liệu.
- Đối thoại truy vấn thường yêu cầu người sử dụng đặc tả một thuộc tính hoặc một nhóm các thuộc tính để phục vụ cho việc tìm kiếm trong cơ sở dữ liệu.
- Sử dụng ngôn ngữ truy vấn đòi hỏi người dùng phải có kinh nghiệm.

2.2.5. Sử dụng form và bảng tính (1) - Form

- Giao diện form được sử dụng chủ yếu để nhập dữ liệu hoặc các ứng dụng khôi phục dữ liệu.
- Giao diện là một form cung cấp các mục thông tin.
- Người sử dụng chọn một mục nào đó để nhập thông tin hoặc hiệu chỉnh nội dung của mục đó.
- Giao diện này dễ học, dễ dùng, đặc biệt thích hợp cho người mới dùng.

2.2.5. Sử dụng form và bảng tính (2) - Form

Đại lý bán vé du lịch

Xin vui lòng nhập các thông tin chi tiết về chuyến đi:

Điểm khởi hành:

Điểm đến:

Đường đi:

Số ghế:

2.2.5. Sử dụng form và bảng tính (3) – Bảng tính

- Bảng tính là một dạng phức tạp hơn của form.
- Bao gồm một lưới các ô, mỗi ô chứa một giá trị nhất định hay một công thức.
- Người sử dụng có thể nhập và thay đổi các giá trị hay các công thức theo một thứ tự bất kỳ.
- Hệ thống có nhiệm vụ duy trì sự nhất quán giữa các giá trị hiển thị và đảm bảo các công thức được thực hiện đúng.
- Bảng tính tạo giao diện linh hoạt, tự nhiên. Người sử dụng được tự do thao tác các dữ liệu tùy ý.

2.2.5. Sử dụng form và bảng tính (4) – Bảng tính

F2		fx =2012- YEAR(E2)				
	A	B	C	D	E	F
1		SỐ TT	Họ Và Tên		Ngày sinh	Tuổi
2		1	Lê Quốc	Bảo	3/22/1990	22
3		2	Nguyễn Văn	Bình	10/23/1989	23
4		3	Lê Văn	Chàng	8/28/1990	22
5		4	Nguyễn Hữu	Công	2/21/1989	23
6		5	Phạm Duy	Đại	3/12/1990	22
7		6	Nguyễn Phương	Điền	9/19/1988	24
8		7	Lê Trung	Đức	9/24/1990	22

2.2.6. Chọn và nhấn (1)

- Đây là kiểu giao diện phổ biến được dùng trong hầu hết các hệ thống multimedia và trình duyệt web.
- Người sử dụng truy cập thông tin bằng cách chọn vị trí truy cập và nhấn chuột.
- Đây là giao diện rất được ưa chuộng trong các trang WWW.

2.2.6. Chọn và nhấn (2)

Yahoo! Việt Nam - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites

Address <http://vn.yahoo.com/>

Xem trước email với Thanh công cụ Yahoo!

YAHOO!
VIỆT NAM

Web Hình ảnh Video Tin tức Blog

Thứ Tư, 11 tháng 1, 2012

CÁC TRANG YAHOO! Sửa

- [Mail](#)
- [Messenger](#)
- [Game](#)
- [Tin Tức](#)
- [Tin Thể Thao](#)
- [Thế Giới Sao](#)
- [Nàng](#)
- [Blog](#)

Vụ cháy xe: Kiểm tra 3.000 mẫu xăng dầu

Tình hình cháy nổ xe ngày càng nghiêm trọng. 3000 mẫu xăng dầu được thu để kiểm định. Nguyên nhân cháy xe? >>

- Đi Tết sếp, 'hàng nóng' cho nhanh
- Hoa xuân xuống phố

Chương 2 – TỔ CHỨC TƯƠNG TÁC

2.1. Mô hình tương tác

2.2. Các dạng tương tác

2.3. Giao diện WIMP

2.4. Điều khiển trực tiếp

2.5. Thiết kế và bố trí màn ảnh

Chương 2 – TỔ CHỨC TƯƠNG TÁC

2.3. Giao diện WIMP

2.3.1. Cửa sổ (Windows)

2.3.2. Biểu tượng (Icons)

2.3.3. Bảng chọn (Menus)

2.3.4. Con trỏ (Pointers)

2.3.5. Các thành phần khác

Giao diện WIMP

- **WIMP**: **W**indows, **I**cons, **M**enus and **P**ointers.
- WIMP thường được gọi là các hệ thống cửa sổ.
- Đây là dạng tương tác mặc định cho phần lớn hệ thống tương tác máy tính đang sử dụng hiện nay.

Ví dụ: Microsoft Windows cho hầu hết các PC

Các thành phần của giao diện WIMP

* Bốn thành phần đặc trưng then chốt:

- Cửa sổ – **Windows**
- Biểu tượng – **Icons**
- Bảng chọn – **Menus**
- Con trỏ – **Pointers**

* Một số thành phần khác:

- Nút bấm – **Buttons**
- Thanh công cụ – **Toolbars**
- Bảng hội thoại – **Dialog boxes**

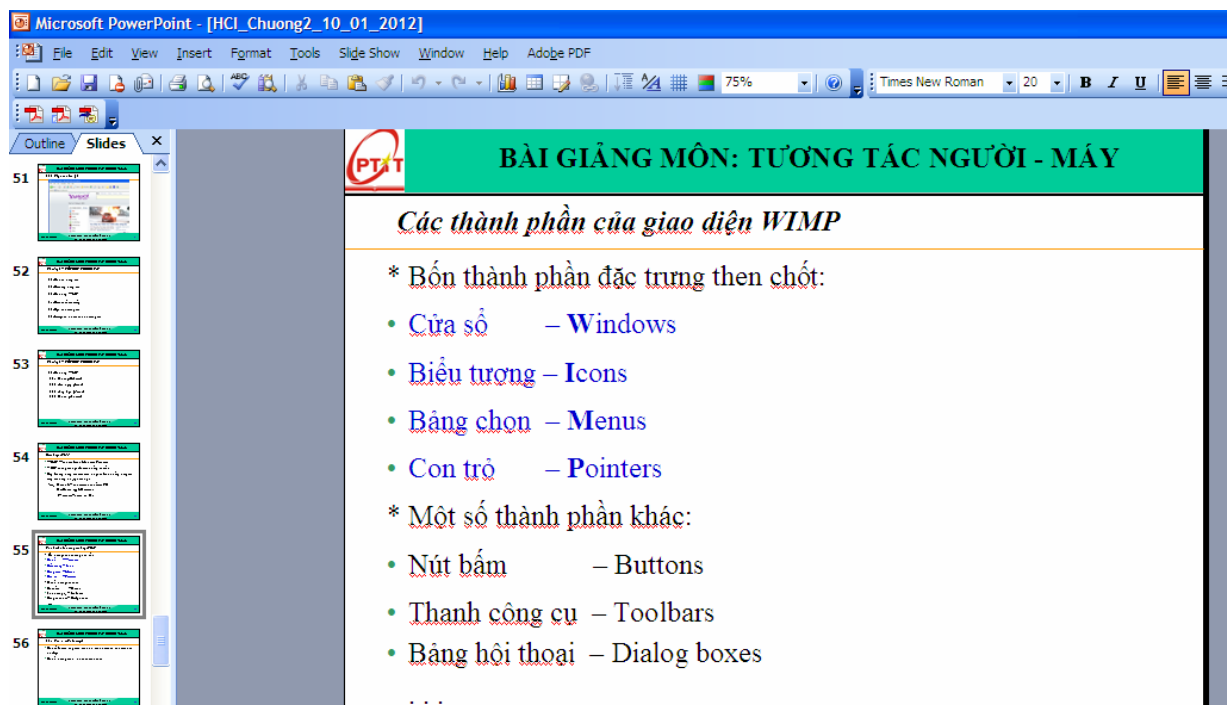
...

Một số ưu điểm của giao diện WIMP

- Có thể hiển thị đồng thời nhiều kiểu thông tin khác nhau, cho phép người sử dụng thực hiện đồng thời nhiều nhiệm vụ.
- Cho phép người sử dụng thực hiện các nhiệm vụ và đối thoại một cách dễ dàng.
- Sử dụng các biểu tượng đồ họa, nút bấm, thanh cuộn, ... làm giảm khối lượng gõ, tăng tính hiệu quả tương tác, đặc biệt với những người mới sử dụng.

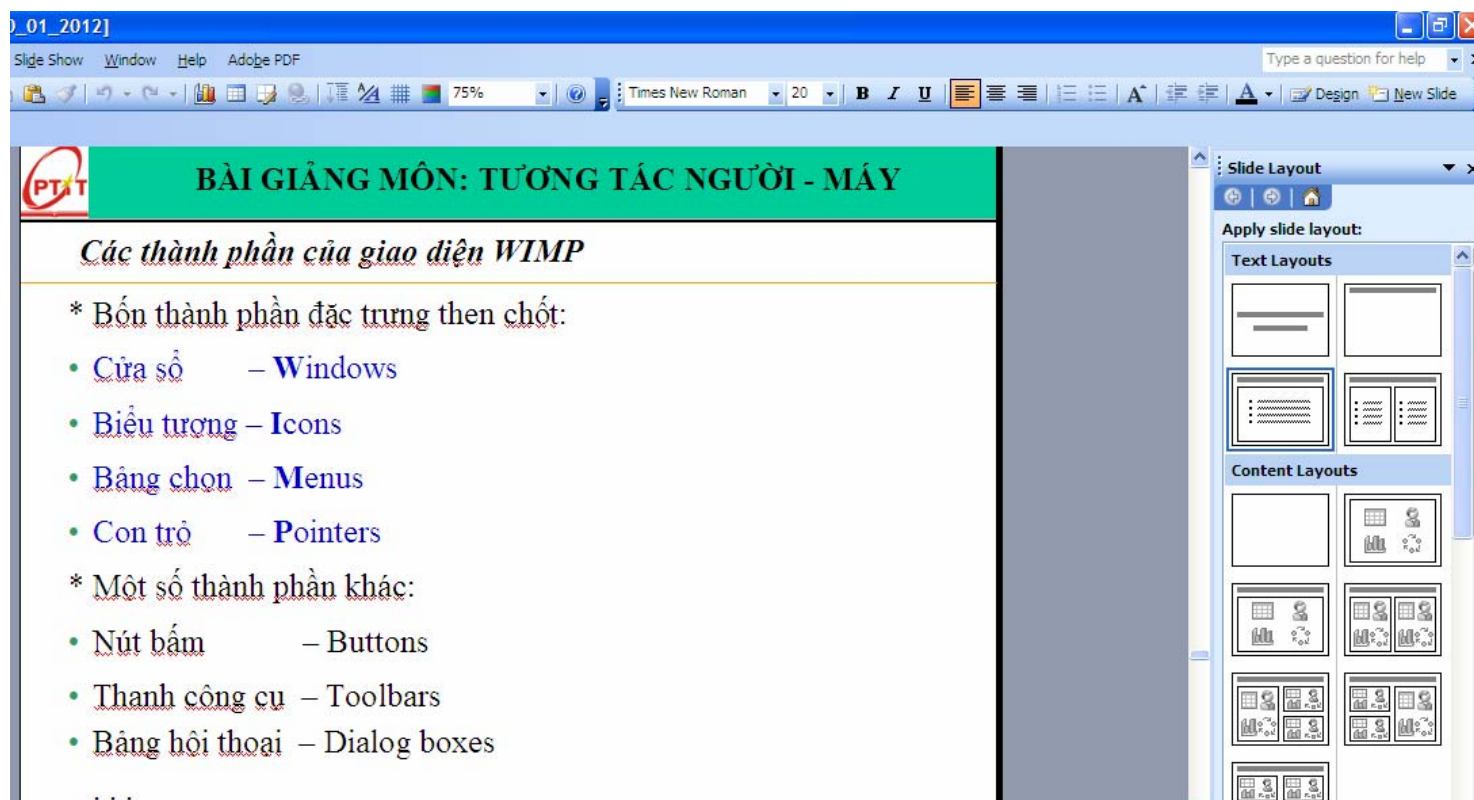
2.3.1. Cửa sổ - Windows (1)

- Cửa sổ là một vùng trên màn hình, được xem như một màn hình độc lập.
- Cửa sổ thường chứa văn bản hoặc đồ họa.



2.3.1. Cửa sổ - Windows (2)

- Có thể di chuyển cửa sổ hay chỉnh lại kích thước cửa sổ theo ý muốn trong giới hạn cho phép.



The screenshot shows a presentation slide titled "Các thành phần của giao diện WIMP" (Components of the WIMP interface). The slide is displayed in a software application window with a menu bar (Slide Show, Window, Help, Adobe PDF) and a toolbar. The slide content is as follows:

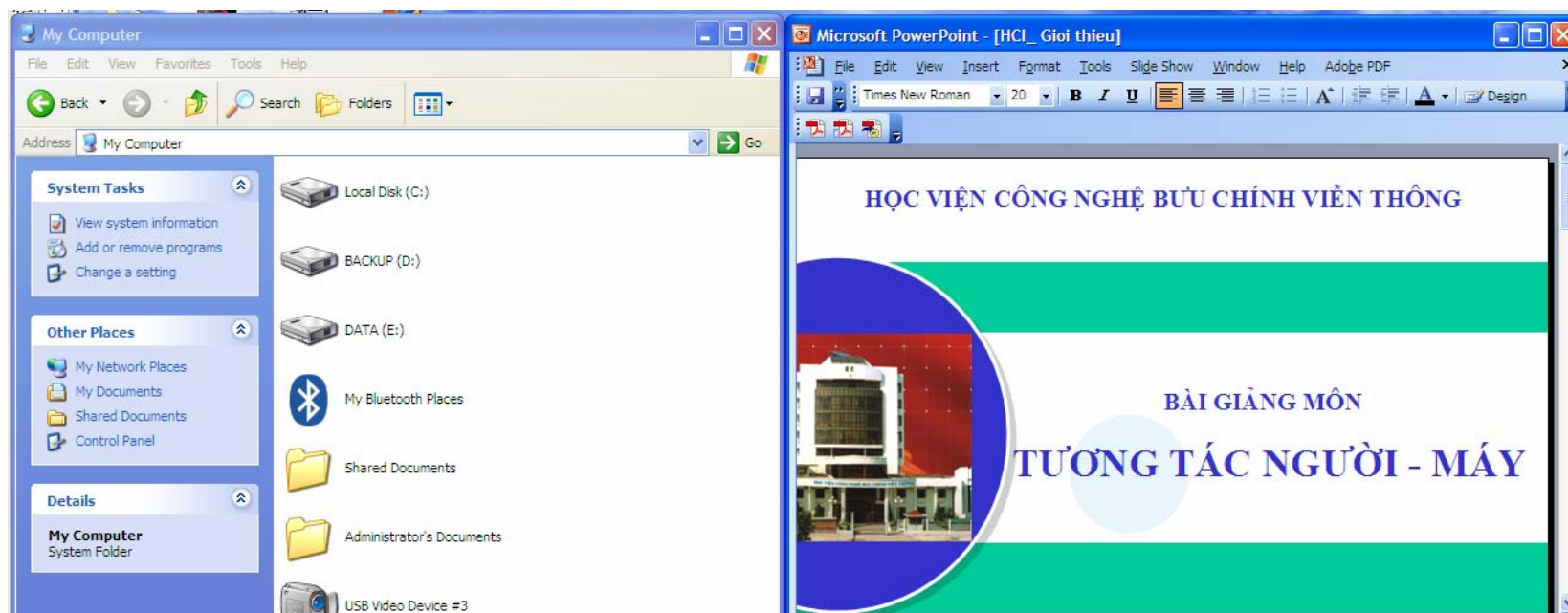
Các thành phần của giao diện WIMP

- * Bốn thành phần đặc trưng then chốt:
 - Cửa sổ – Windows
 - Biểu tượng – Icons
 - Bảng chọn – Menus
 - Con trỏ – Pointers
- * Một số thành phần khác:
 - Nút bấm – Buttons
 - Thanh công cụ – Toolbars
 - Bảng hội thoại – Dialog boxes

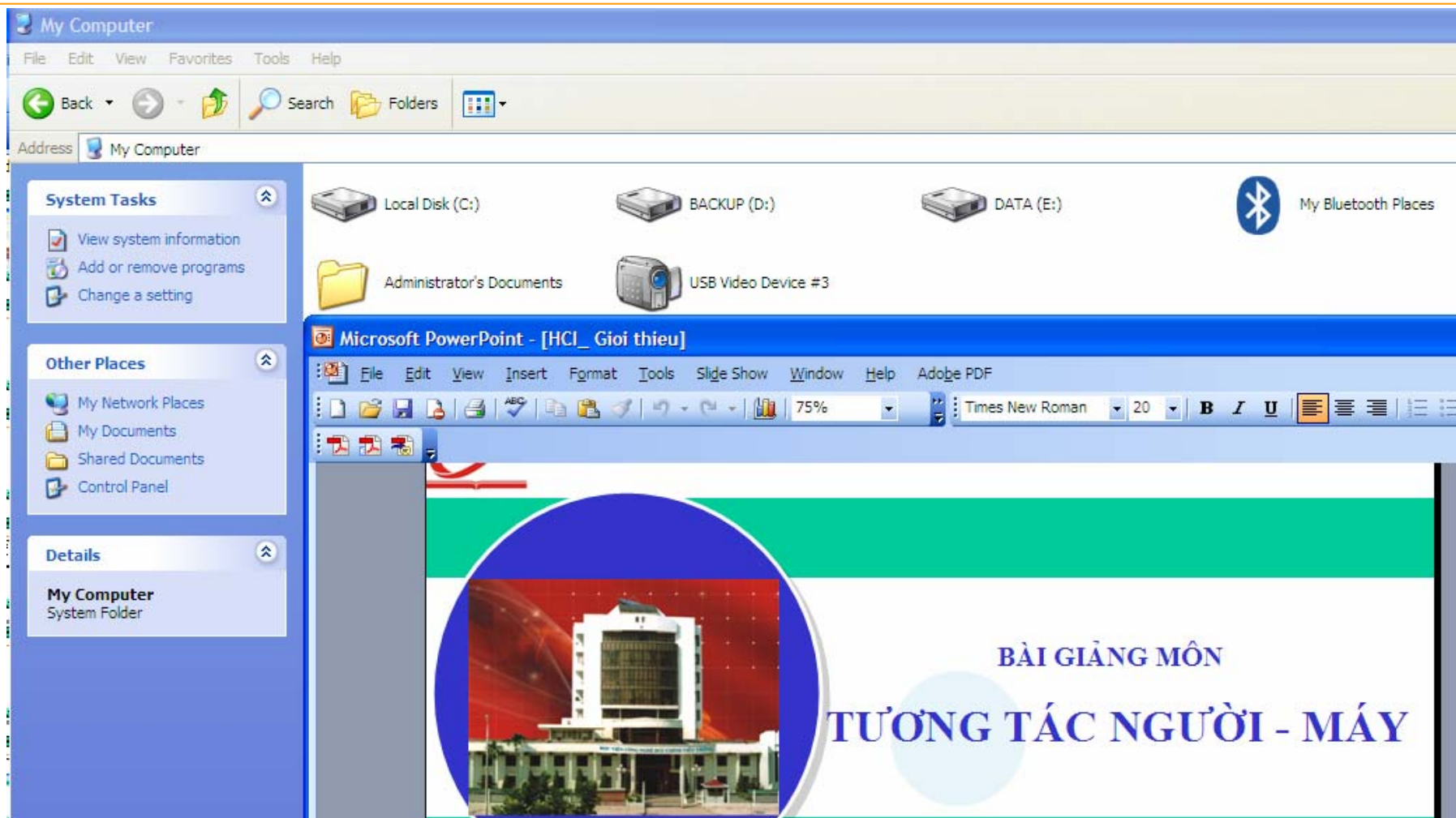
2.3.1. Cửa sổ - Windows (3)

- Có thể có nhiều hơn một cửa sổ trên màn hình cùng lúc, cho phép các công việc khác nhau được xuất hiện cùng lúc.
- Sắp xếp nhiều cửa sổ trên màn hình:
 - Xếp kê: các cửa sổ được sắp xếp cạnh nhau.
 - Xếp tầng: cửa sổ mới được đặt hơi dịch sang phải và phía dưới cửa sổ trước.
- Người sử dụng có thể chuyển đổi giữa các cửa sổ để thao tác trên các cửa sổ khác nhau.

2.3.1. Cửa sổ - Windows (4) – Cửa sổ xếp kề

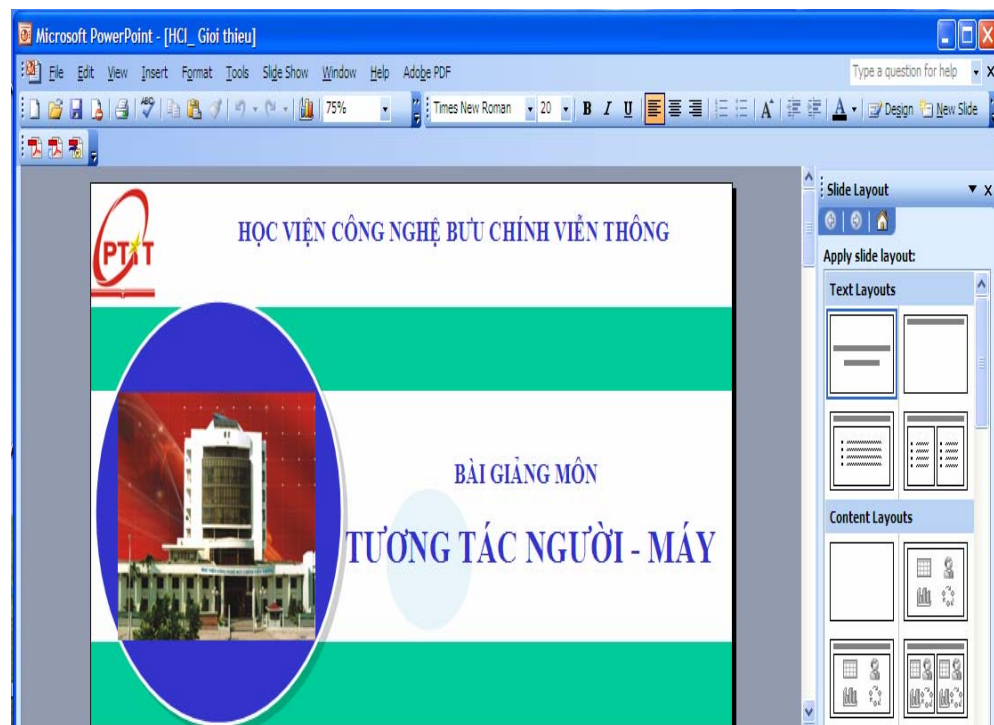


2.3.1. Cửa sổ - Windows (5) – Cửa sổ xếp tầng



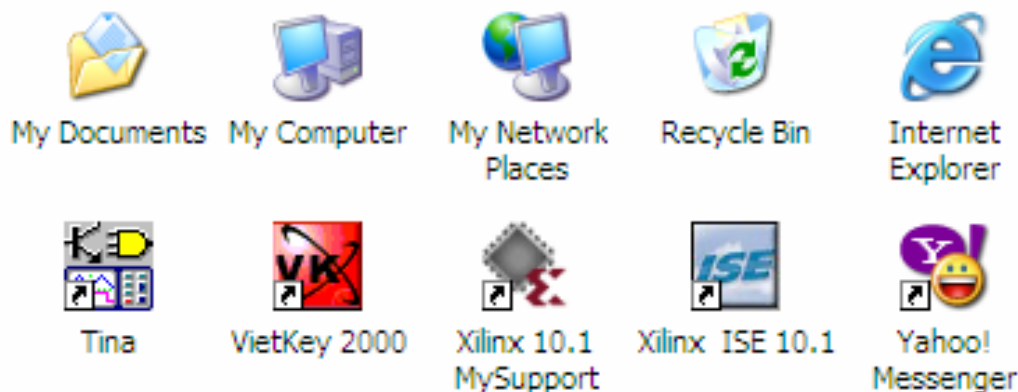
2.3.1. Cửa sổ - Windows (6) – Một số thành phần kết hợp

- Thanh cuộn: giúp tăng sự linh hoạt, cho phép người sử dụng di chuyển nội dung cửa sổ lên xuống, sang trái sang phải.
- Thanh tiêu đề: giúp nhận dạng cửa sổ đang thao tác.
- Các nút bấm: giúp chỉnh kích cỡ, đóng hoặc phóng to cửa sổ.



2.3.2. Biểu tượng – Icons

- Cửa sổ có thể được đóng lại hoặc thu nhỏ lại thành biểu tượng.
- Cho phép hiện đồng thời nhiều biểu tượng trên màn hình.
- Nhấn vào biểu tượng: mở rộng cửa sổ tương ứng tới kích thước lớn nhất.
- Có thể sử dụng biểu tượng để thể hiện một số chức năng của hệ thống (như thùng rác, mạng máy tính,...).



2.3.3. Bảng chọn – Menus (1)

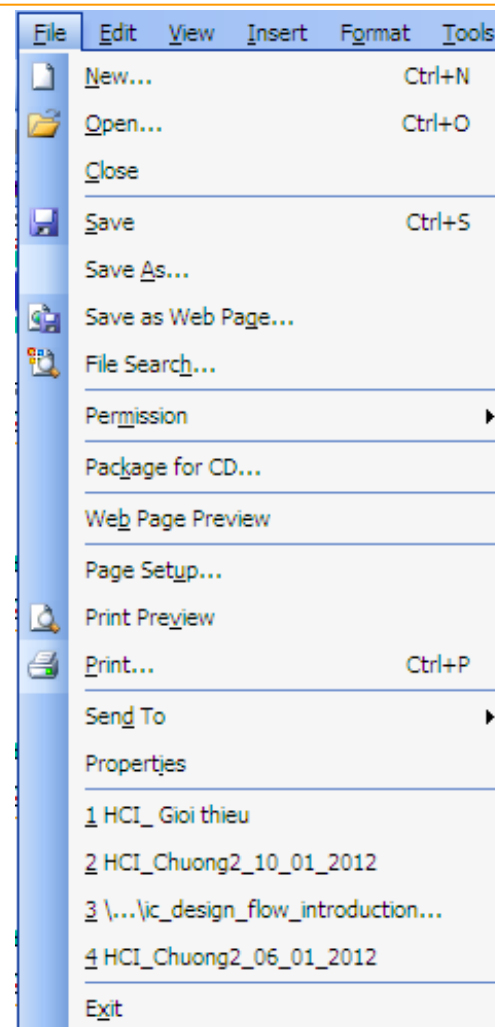
- Menu là đặc trưng của các hệ thống cửa sổ, là kỹ thuật tương tác phổ biến dùng cho cả các hệ thống không cửa sổ.
- Đưa ra một lựa chọn các thao tác hay dịch vụ có thể được hệ thống trình diễn.
- Tên được sử dụng trong menu nên có nhiều ý nghĩa và cung cấp nhiều thông tin.

2.3.3. Bảng chọn – Menus (2)

- Con trỏ được sử dụng để cho biết yêu cầu lựa chọn.
- Khi di chuyển con trỏ tới vị trí mục chọn trong menu thì mục này thường được làm nổi bật nhất (sử dụng hình ảnh hoặc màu sắc khác đi).
- Người sử dụng di và nhấn chuột hoặc nhấn một vài phím đặc biệt để lựa chọn một mục nào đó.
- Menu thường không hiệu quả khi có quá nhiều mục lựa chọn.
- Sử dụng bảng chọn nhiều tầng, chỉ hiện ra những lựa chọn hay dùng, ẩn đi những lựa chọn ít dùng để tăng hiệu quả sử dụng.

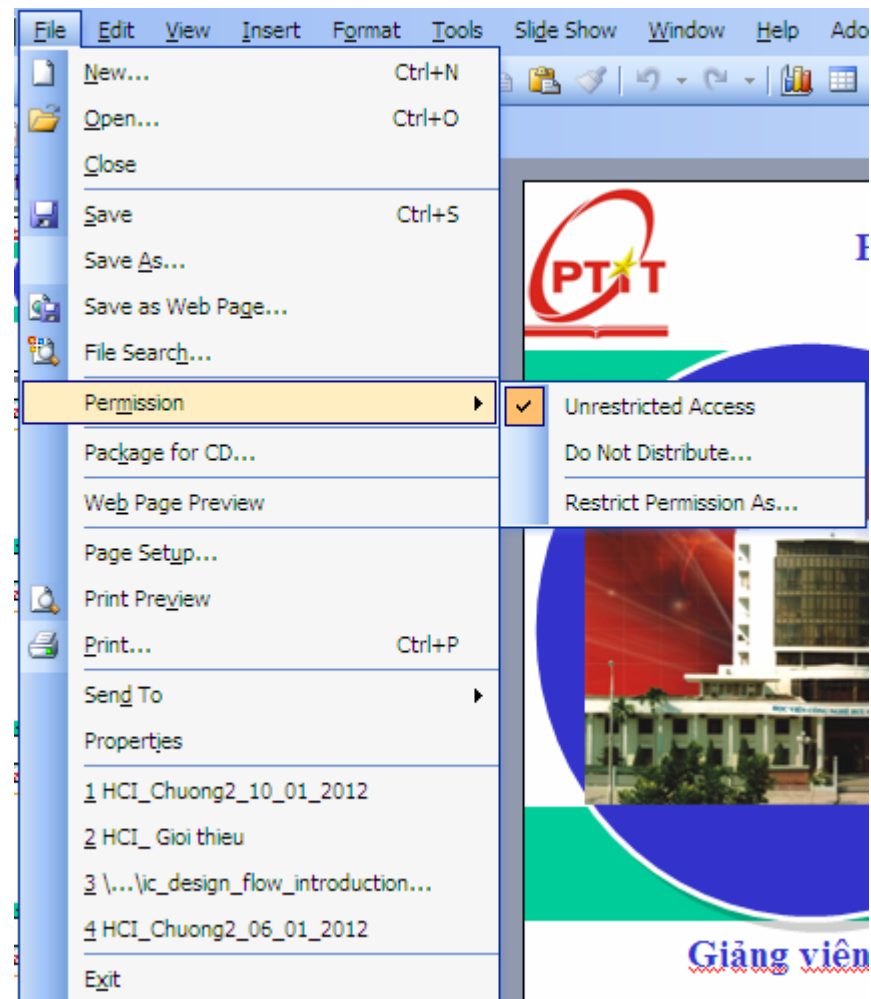
2.3.3. Bảng chọn – Menus (3) – Bảng chọn đổ xuống (pull – down)

- Các menu chính thường được đặt trên đỉnh màn hình.
- Di chuyển con trỏ đến vùng tiêu đề tương ứng và nhấn chuột, các lựa chọn sẽ xuất hiện.



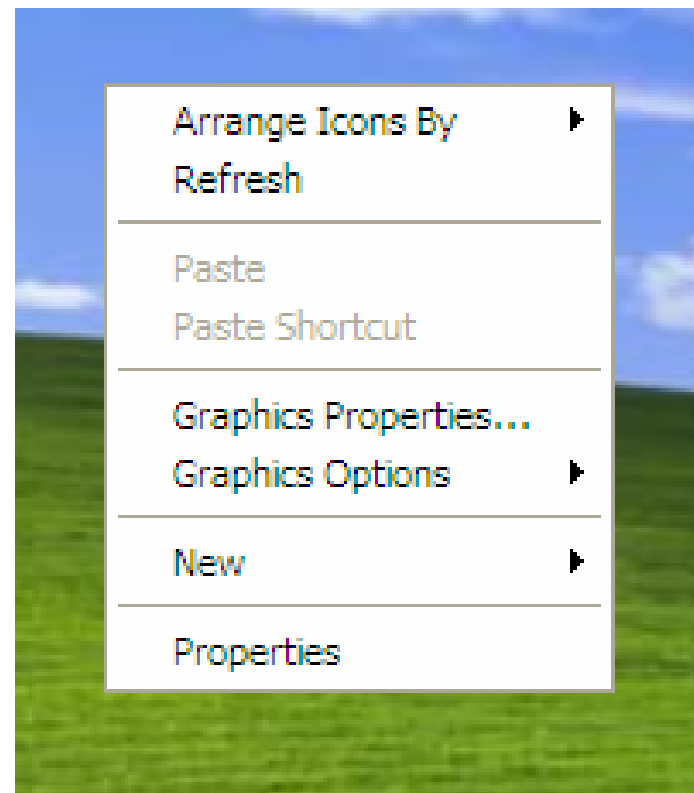
2.3.3. Bảng chọn – Menus (4) – Bảng chọn rơi xuống (drop – down)

- Chỉ cần di chuyển con trỏ đến vùng tiêu đề tương ứng, các lựa chọn sẽ xuất hiện.



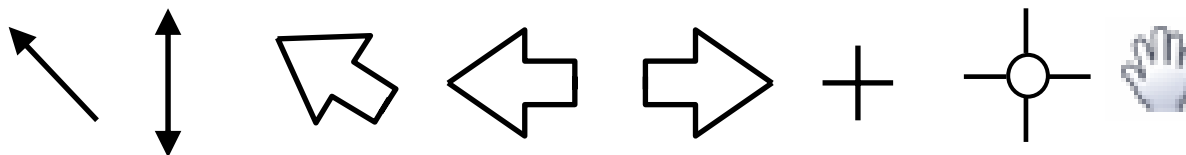
2.3.3. Bảng chọn – Menus (5) – Bảng chọn bật lên (pop – up)

- Menu chính bị ẩn, khi có yêu cầu nó sẽ xuất hiện bất ngờ trên màn hình.
- Nhấn chuột phải trên màn hình để làm xuất hiện các lựa chọn.



2.3.4. Con trỏ - Pointers

- Con trỏ là thành phần quan trọng, vì WIMP dùng chủ yếu định vị và lựa chọn.
- Các thiết bị đầu vào cung cấp khả năng định vị và lựa chọn: chuột (phổ biến nhất), joysticks, trackballs,...
- Điểm trỏ: là vị trí mà con trỏ trỏ vào. Các con trỏ phải được thiết kế với điểm trỏ rõ ràng.
- Một số dạng con trỏ:



2.3.5. Các thành phần khác (1) – Nút bấm

- Bấm nút để thực hiện một lệnh.
- Nút bấm luôn xuất hiện trong cửa sổ (khác với menu).
- Ba loại nút bấm:
 - Nút bấm bình thường
 - Nút bấm chọn
 - Hộp kiểm tra

2.3.5. Các thành phần khác (2) – Thanh công cụ

- Là nhóm các nút bấm hiển thị thông qua các biểu tượng.
- Cho phép người sử dụng tùy chỉnh.

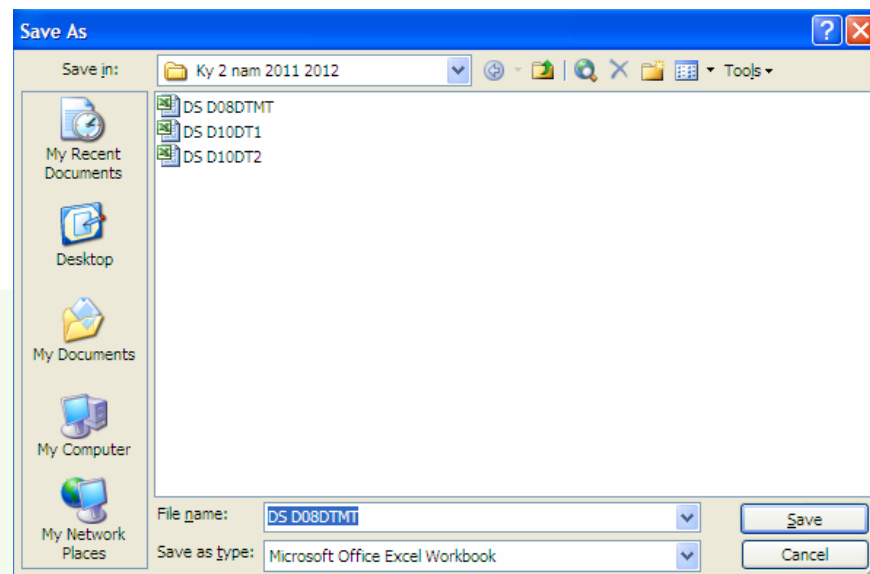
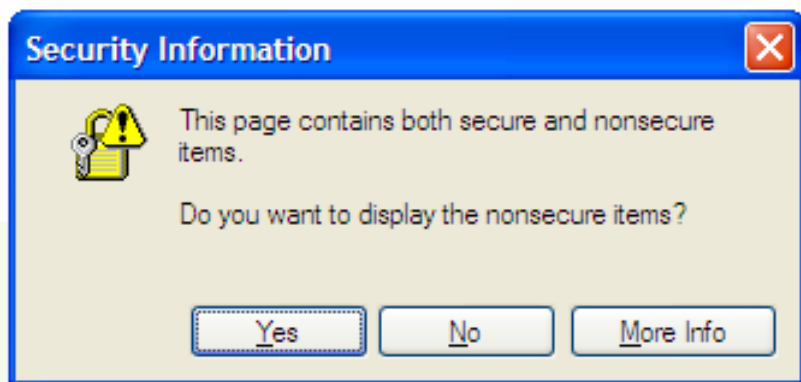


2.3.5. Các thành phần khác (3) – Hộp hội thoại

- Thu hút sự chú ý của người dùng.

Ví dụ: báo lỗi, cảnh báo,...

- Có thể là hệ thống hội thoại con cho một nhiệm vụ nhất định.



Chương 2 – TỔ CHỨC TƯƠNG TÁC

2.1. Mô hình tương tác

2.2. Các dạng tương tác

2.3. Giao diện WIMP

2.4. Điều khiển trực tiếp

2.5. Thiết kế và bố trí màn ảnh

Điều khiển trực tiếp

- Sử dụng phổ biến trong thiết kế các hệ thống văn phòng.
- Người dùng có thể giao tiếp thông qua bất cứ biểu tượng có sẵn nào trên màn hình.

Ví dụ: gấp và thả các đối tượng để di chuyển vị trí
gấp và thả đối tượng vào thùng rác để xóa

- Ưu điểm: đơn giản, tự nhiên, dễ dùng.
- Nhược điểm: đôi khi có sự nhập nhằng.
- Điều khiển trực tiếp hầu như không được sử dụng trong các thiết kế công nghiệp.

Chương 2 – TỔ CHỨC TƯƠNG TÁC

2.1. Mô hình tương tác

2.2. Các dạng tương tác

2.3. Giao diện WIMP

2.4. Điều khiển trực tiếp

2.5. Thiết kế và bố trí màn ảnh

Chương 2 – TỔ CHỨC TƯƠNG TÁC

2.5. Thiết kế và bố trí màn ảnh

2.5.1. *Hiển thị thông tin*

2.5.2. *Nhập thông tin*

2.5.3. *Thẩm mỹ học và tính hữu dụng*

2.5.1. *Hiển thị thông tin (1)*

- Thông tin phải được hiển thị đầy đủ, rõ ràng, dễ hiểu để thỏa mãn nhu cầu của người dùng.
- Các cách hiển thị thông tin: văn bản, hình ảnh và âm thanh, kích thước, màu sắc, độ phân giải ...
- Hiển thị thông tin phụ thuộc vào các yếu tố:
 - Loại thông tin: chữ, số, bản đồ, bảng, hình vẽ,...
 - Công nghệ hiện có để trình diễn: biểu diễn chữ, hình vẽ, đồ họa,...
 - Mục đích của thông tin (yếu tố quan trọng nhất).

2.5.1. *Hiển thị thông tin (2) – Một số hướng dẫn*

- Chỉ hiển thị thông tin có liên quan đến ngữ cảnh hiện tại.
- Sử dụng định dạng trình bày cho phép dễ dàng thu nhận thông tin.
Ví dụ: sử dụng đồ thị hay sơ đồ thay cho các bảng biểu.
- Sử dụng tên gọi nhất quán, cách viết tắt chuẩn.
- Cho phép người dùng duy trì ngữ cảnh trực quan. Nếu thay đổi tỉ lệ hiển thị thì nên hiển thị hình ảnh gốc thường xuyên dưới dạng rút gọn ở góc màn hình, để người dùng biết được vị trí tương đối của phần hình ảnh đang được xét trong hình ảnh gốc.

2.5.1. *Hiển thị thông tin (3) – Một số hướng dẫn*

- Đưa ra thông báo lỗi có nghĩa.
- Sử dụng chữ hoa, chữ thường, căn lề, gộp nhóm văn bản để hiển thị thông tin được rõ ràng.
- Sử dụng hiệu quả vùng hiển thị có sẵn trên màn hình. Khi dùng nhiều cửa sổ, nên có sẵn không gian để chỉ ra một phần cho từng cửa sổ.
- Nên lựa chọn cài đặt kích cỡ màn hình phù hợp với kiểu ứng dụng.

2.5.1. *Hiển thị thông tin (4) – Ví dụ*

- Không nên sử dụng quá nhiều màu trên màn hình. Khi in ra giấy, màn hình phải nhìn rõ được.
- Tầm quan trọng khi căn lề:
 - Đối với chữ: nên căn theo lề bên trái, vì chúng ta đọc từ trái sang phải.
 - Đối với số: nên căn theo lề bên phải, hoặc căn lề từ dấu phẩy để người dùng biết được độ lớn của số.

2.5.1. *Hiển thị thông tin (5) – Ví dụ*

Cách hiển thị tốt

Vì căn lề từ trái sang phải

Cách hiển thị chưa tốt

Vì căn lề từ phải sang trái

12 345

519 391

3 820 866

12 345

519 391

3 820 866

1234.5

519.391

38.20866

1234.5

519.391

38.20866

2.5.2. Nhập thông tin (1)

- Cần tối thiểu số hành động nhập thông tin mà người dùng phải thực hiện.
- Duy trì sự nhất quán giữa thông tin nhập vào và thông tin hiển thị.
- Tương tác mềm dẻo, cho phép người dùng lựa chọn thiết bị nhập thông tin (dùng bàn phím hay chuột, ...)
- Người dùng có thể bỏ qua các hành động không cần thiết, thay đổi trật tự các hành động yêu cầu.
- Hệ thống có khả năng cung cấp hỗ trợ cho mọi hành động nhập thông tin.
- Bố trí các mục nhập thông tin một cách hợp lý, thuận tiện.

2.5.2. Nhập thông tin (2) – Ví dụ

- Khi các mục nhập thông tin và các nhãn đi kèm có độ dài khác nhau, cần bố trí sao cho hợp lý.

Tên	<input type="text"/>
Ngày sinh	<input type="text"/>

Tên	<input type="text"/>
Ngày sinh	<input type="text"/>

2.5.3. Thẩm mỹ học và tính hữu dụng

- Một giao diện đẹp không phải luôn đồng nghĩa với một giao diện tốt.
- Lý tưởng nhất là một giao diện vừa đẹp, vừa hữu dụng. Tuy nhiên, khi phải lựa chọn giữa hai tiêu chí, tính hữu dụng nên được đặt lên trước tính thẩm mỹ.
- Giao diện hữu dụng phải cho người dùng biết được có thể thao tác với cái gì (nút bấm, bảng chọn, ...) và không thể thao tác với cái gì (nhãn, thông tin,...).
- Giao diện nên được thiết kế theo các mẫu chuẩn (thiết kế các nút bấm giống nhau, bảng chọn giống nhau,...).

Chương 3 – THIẾT KẾ TƯƠNG TÁC NGƯỜI - MÁY

3.1. Các nguyên tắc khi thiết kế tương tác người - máy

3.2. Quy trình thiết kế

3.3. Mô hình người dùng

3.4. Phân cấp mục tiêu và nhiệm vụ

3.5. Mô hình ngôn ngữ

3.6. Mô hình mức vật lý

Chương 3 – THIẾT KẾ TƯƠNG TÁC NGƯỜI - MÁY

3.1. Các nguyên tắc khi thiết kế tương tác người - máy

3.1.1. *Tổng quan về thiết kế tương tác người - máy*

3.1.2. *Các mô thức cho tính dùng được*

3.1.3. *Các nguyên tắc cho tính dùng được*

3.1.1. Tổng quan về thiết kế tương tác người – máy (1)

- Mục tiêu của thiết kế hệ thống tương tác: đảm bảo tính dùng được tối đa.
- Tính dùng được: hệ thống cho phép người dùng thực hiện các nhiệm vụ một cách hiệu quả.
- Thiết kế các hệ thống tương tác cần trả lời hai câu hỏi:
 1. Làm thế nào để đảm bảo tính dùng được của hệ thống tương tác được phát triển?
 2. Làm thế nào để chứng minh hay đánh giá tính dùng được của một hệ thống tương tác?

3.1.1. Tổng quan về thiết kế tương tác người – máy (2)

Hai cách tiếp cận để trả lời hai câu hỏi trên:

- Cách thứ nhất: thông qua các ví dụ.
 - Xem xét các hệ thống tương tác đã được thiết kế thành công.
 - Lấy các ví dụ đó làm mô thức cho việc phát triển các hệ thống tương tác trong tương lai.
- Cách thứ hai: mang tính lý thuyết.
 - Dựa trên các tri thức về tâm lý học, lý thuyết tính toán, xã hội học,... đưa ra các nguyên tắc cho việc tương tác hiệu quả.
 - Lấy các nguyên tắc này để điều khiển thiết kế và đánh giá sản phẩm.

3.1.2. Các mô thức cho tính dùng được (1)

- Mô thức cung cấp cho con người niềm tin là các hệ thống hiện tại có tính dùng được cao hơn các hệ thống trước đây, và các hệ thống tương lai lại tốt hơn các hệ thống hiện tại.
- Để làm tăng tính dùng được, cần kết hợp tính sáng tạo của con người và các công nghệ hiện đại.
- Mô thức phụ thuộc vào công nghệ và bản chất các ứng dụng.
- Có 14 mô thức, chúng không loại trừ mà bổ sung, hỗ trợ lẫn nhau.

3.1.2. Các mô thức cho tính dùng được (2) – 14 mô thức

- Phân chia thời gian (time sharing)
- Các thiết bị hiển thị quan sát (Video Display Units – VDU)
- Các công cụ lập trình (Programming Toolkits)
- Máy tính cá nhân (Personal Computer)
- Hệ thống cửa sổ và giao diện WIMP
- Phép ẩn dụ (Metaphor)
- Điều khiển trực tiếp (Direct manipulation)

3.1.2. Các mô thức cho tính dùng được (3) – 14 mô thức

- Ngôn ngữ và hành động (language versus action)
- Siêu văn bản (Hyper Text)
- Đa phương thức (Multi – modality)
- WEB
- Giao tiếp dựa vào tác tử (Agents based interface)
- Hợp tác có máy tính hỗ trợ (Computer – supported cooperative work)
- Tin học mọi nơi (ubiquitous computing)

3.1.2.1. Phân chia thời gian

- Thập kỷ 1940 – 1950 là thời gian đánh dấu sự bùng nổ về phần cứng trong ngành tin học. Sự ra đời của đèn điện tử, transistor,... làm tăng khả năng tính toán lên gấp nhiều lần.
- Tuy nhiên, mỗi máy tính chỉ có thể phục vụ một người làm một nhiệm vụ tại một thời điểm, gây lãng phí khả năng tính toán.
- Thập kỷ 1960, kỹ thuật phân chia thời gian ra đời, cho phép một máy tính có thể hỗ trợ đa người dùng.

3.1.2.2. Các thiết bị hiển thị quan sát

- Trước đây, máy tính nhập hay hiển thị thông tin thông qua bìa đục lỗ (các bit 0/1).
- Thập kỷ 1950, các thiết bị hiển thị video ra đời.
- VDU cho phép hiển thị nhiều loại thông tin (chữ, hình ảnh,...) thay vì chỉ có các bit.
- VDU giúp nâng cao khả năng tương tác giữa con người và máy tính.

3.1.2.3. Các công cụ lập trình

- Ý tưởng dùng máy tính làm phương tiện giúp đỡ con người giải quyết vấn đề do Douglas Engelbart đưa ra vào những năm 1950.
- Để có được các thiết bị tính toán trợ giúp khả năng giải quyết vấn đề, phải cung cấp các ý tưởng đúng.
- Ý tưởng về việc xây dựng các thành phần của hệ thống tính toán được gọi là quá trình khởi động, và được sử dụng mở rộng trong mọi tính toán.
- Có thể tổ hợp các công cụ lập trình nhỏ để tạo ra các công cụ lớn hơn.

3.1.2.4. Máy tính cá nhân

- Trước những năm 1970, các máy tính đều có kích thước lớn, được đặt ở các phòng thí nghiệm, nhiều người sử dụng chung một chiếc máy tính.
- Đến những năm 1970, bắt đầu xuất hiện các máy tính cá nhân với kích thước nhỏ hơn, giúp cho việc sử dụng các bộ công cụ lập trình trở nên thuận tiện.
- Hiện nay, cùng với sự phát triển của công nghệ, các máy tính cá nhân ngày càng nhỏ gọn. Rất khó để phân biệt máy tính cá nhân, máy chủ hay các máy trạm.

3.1.2.5. Hệ thống cửa sổ và giao diện WIMP

- Con người có khả năng suy nghĩ về nhiều vấn đề tại một thời điểm. Con người thường dừng công việc đang làm để chuyển sang các công việc liên quan khác.
- Nếu hệ thống máy tính chỉ cho phép người sử dụng chạy duy nhất một chương trình từ đầu đến cuối thì không phù hợp với dòng suy nghĩ của người dùng.
- Máy tính cá nhân phải mềm dẻo trong khả năng thay đổi các chủ đề như con người, đồng thời phải đảm bảo tách biệt một cách vật lý các biểu diễn đối thoại người dùng – máy tính khác nhau.
- Hệ thống cửa sổ và giao diện WIMP là cơ chế tương tác quen thuộc.

3.1.2.6. Phép ẩn dụ

- Đưa các khái niệm, hình ảnh trong cuộc sống vào máy tính, giúp người dùng nhanh chóng quen với các khái niệm mới.

Ví dụ: khái niệm thùng rác, bảng tính,...

- Phép ẩn dụ tốt phải làm tăng tính thân thuộc giữa người dùng và các ứng dụng máy tính.
- Đôi khi các khái niệm, hình ảnh như nhau song hành vi lại khác nhau, gây ra các nhầm lẫn khi thao tác trên máy tính.

3.1.2.7. Điều khiển trực tiếp (1) - Các đặc trưng chính

- Các đối tượng đang được quan tâm phải nhìn thấy được.
- Cho phép thực hiện các hành động nối tiếp nhau trên giao diện với phản hồi nhanh chóng.
- Cho phép hủy bỏ thao tác vừa thực hiện (undo), giúp người dùng thử nghiệm các thao tác mà không phải chịu hậu quả nặng nề.
- Mọi hành động có cú pháp chính xác, do đó mỗi hành động của người dùng đều là các thao tác hợp lệ.
- Ngôn ngữ dòng lệnh phức tạp được thay thế bằng các thao tác trực tiếp trên các đối tượng nhìn thấy được.

3.1.2.7. Điều khiển trực tiếp (2)

- Giao diện điều khiển trực tiếp là giao diện phổ biến cho các máy tính cá nhân.
- Giao diện điều khiển trực tiếp không có sự phân biệt rõ ràng giữa đầu vào và đầu ra.
- Điều khiển trực tiếp phản ánh tính trực quan trong tương tác, còn được gọi là “WYSIWYG” (What You See Is What You Get). Trong đó, biểu diễn trên máy tính và sản phẩm cuối cùng nhận được có sự khác nhau rất nhỏ.
- Điều khiển trực tiếp giúp việc thực hiện một số nhiệm vụ được dễ dàng và chính xác.

3.1.2.8. Ngôn ngữ và hành động

- Giao tiếp kiểu điều khiển trực tiếp làm cho một số nhiệm vụ trở nên khó khăn hơn, thậm chí không thể thực hiện.
- Với những nhiệm vụ phức tạp, nếu cố tình thiết kế thao tác thông qua các hành động trực tiếp sẽ làm cho người dùng khó sử dụng hệ thống hơn.
- Người dùng giao tiếp bằng ngôn ngữ gián tiếp thay vì hành động trực tiếp.

Ví dụ: Khi thao tác trên nhiều tệp cùng một lúc: Để đổi tên tất cả các tệp từ file .dat sang file .txt thì sử dụng câu lệnh sẽ đơn giản hơn sử dụng hành động.

3.1.2.9. Siêu văn bản

- Siêu văn bản là một khái niệm khá quen thuộc, đặc biệt trong môi trường WEB.
- Là kỹ thuật cho phép truy nhập một cách ngẫu nhiên vào một văn bản.
- Văn bản được hiểu là một khái niệm mở rộng, không phụ thuộc vào không gian lưu trữ vật lý, một phần văn bản có thể nằm trên một tệp khác trên cùng một máy, trên một máy khác cùng mạng hoặc trên một website.
- Thông tin trong siêu văn bản được truy cập phi tuyến tính (truy cập tuyến tính: truy cập từ đầu đến cuối).

3.1.2.10. Đa phương thức

- Hệ thống tương tác thông thường: gồm đầu vào (bàn phím, chuột) và đầu ra (màn hình hiển thị). Các thiết bị vào ra này được gọi là các kênh liên lạc của hệ thống.
- Hệ thống đa phương thức: sử dụng nhiều kênh liên lạc khác nhau với con người.
- Mỗi kênh liên lạc được gọi là một phương thức tương tác.
- Mọi hệ thống tương tác có thể coi như đa phương thức với người dùng luôn sử dụng kênh thị giác và xúc giác để điều khiển máy tính.

3.1.2.11. WEB

- WEB là một trong những phát triển quan trọng nhất của ngành CNTT.
- Nội dung WEB: văn bản, ảnh màu, phim, âm thanh, các siêu liên kết để liên kết đến các trang khác.
- Giao tiếp đồ họa trong trình duyệt WEB rất trực quan và dễ dùng, cho phép người dùng truy nhập các thông tin đa phương tiện một cách dễ dàng chỉ bằng thao tác di chuột và nhấn phím.
- Lướt WEB trở thành một hoạt động chính của con người trong cả công việc lẫn khi giải trí.

3.1.2.12. *Giao tiếp dựa vào tác tử*

- Tác tử: là đơn vị (người, đại lý) đại diện cho một người nào đó.
Ví dụ: đại lý bán và mua hàng, đại lý sách, hãng du lịch,...
 - Tác tử phần mềm: hoạt động nhân danh người dùng con người trong thế giới điện tử.
 - Tác tử điện tử có thể thực hiện các nhiệm vụ lặp, quan sát, trả lời các sự kiện khi người dùng vắng mặt,...
- Ví dụ: tác tử thư điện tử thực hiện lọc thư cho người dùng; trình duyệt WEB tìm các trang WEB hay các tư liệu trên trang WEB; tác tử tính toán trong Excel,...

3.1.2.13. Hợp tác có máy tính hỗ trợ

- Sự ra đời của mạng máy tính vào khoảng thập kỷ 1960 là một trong những phát triển vượt bậc trong ngành CNTT, cho phép truyền thông giữa các máy tính riêng biệt.
- Tương tác giữa con người với máy tính không đơn thuần là giao tiếp giữa một người với một máy tính, mà là giữa nhiều người với nhau và với nhiều máy tính.

Ví dụ: email là một ví dụ của mô hình hợp tác có máy tính hỗ trợ, là sự trao đổi thư từ giữa nhiều người với nhau, nhưng thông qua hệ thống máy tính.

3.1.2.14. Tin học mọi nơi

- Ý tưởng này được phát triển vào khoảng thập kỷ 1980, với mục đích:
 - Đưa việc tương tác giữa con người với máy tính ra khỏi bàn máy tính, đi vào mọi nơi trong cuộc sống.
 - Làm cho con người không còn nhận ra máy tính trong cuộc sống hàng ngày nữa.
- Chế tạo máy tính dưới mọi kích thước là điều quan trọng.
Ví dụ: đồng hồ, lò vi sóng, điện thoại di động,...
- Nhiều công nghệ đã và đang được phát triển: mạng không dây, nhận dạng tiếng nói, định vị toàn cầu, ...

3.1.3. Các nguyên tắc cho tính dùng được

- 5 nguyên tắc của Nielsen (1993):
 - Tính dễ học.
 - Tính hiệu quả.
 - Tính dễ nhớ.
 - Tính dự đoán lỗi.
 - Đáp ứng tính chủ quan.
- Tập trung thành 3 tiêu chí chính:
 - Tính dễ học (learnability).
 - Tính mềm dẻo (flexibility).
 - Tính vững chắc (robustness).

3.1.3.1. Tính dễ học (1)

- Là một đặc trưng quan trọng của hệ thống tương tác, cho phép người dùng (kể cả mới và thiếu kinh nghiệm) có thể sử dụng đạt hiệu quả tối đa.
- Các đặc trưng chính của tính dễ học:
 - Tính dự đoán
 - Tính tổng hợp
 - Tính thân thiện
 - Tính khái quát
 - Tính nhất quán

3.1.3.1. Tính dễ học (2)

* *Tính dự đoán:*

- Người dùng có thể xác định được các tương tác mới và đoán được kết quả tương tác dựa vào tri thức đã biết.
- Tính dự đoán được chia thành nhiều cấp độ, từ mức thấp (yêu cầu tri thức hạn chế) đến mức cao hơn (yêu cầu tri thức tăng cường).
- Tính dự đoán thể hiện khả năng của người dùng, nhằm xác định tác động của các thao tác với hệ thống.
- Hệ thống nên hỗ trợ các suy luận hay dự đoán này bằng cách luôn luôn đưa ra những thông tin phản hồi nhất quán.

3.1.3.1. Tính dễ học (3)

* Tính tổng hợp:

- Tính tổng hợp là khả năng người dùng hình thành một mô hình về hành vi của hệ thống.
- Khi trạng thái của hệ thống thay đổi, người dùng phải dễ dàng nhận biết được sự thay đổi này.

- Tính tổng hợp tốt: sự nhận biết này diễn ra ngay lập tức.

- Tính tổng hợp chưa tốt: sự nhận biết này xuất hiện một cách ngẫu nhiên sau khi có thông báo về sự thay đổi trạng thái.

Ví dụ: hệ thống điều khiển trực tiếp có tính tổng hợp tốt. Khi người dùng chuyển một tệp từ thư mục này sang thư mục khác bằng thao tác gấp thả, nhận ra ngay sự thay đổi.

3.1.3.1. Tính dễ học (4)

* *Tính thân thiện:*

- Người dùng thường vận dụng những kinh nghiệm của mình trong một lĩnh vực nào đó để quyết định sẽ phải làm gì trong các tình huống mới.
- Các hệ thống nên vận dụng những kiến thức theo quy luật, để người dùng không phải suy nghĩ quá nhiều trước một tình huống mới.
- Tính thân thiện của một hệ thống được đo bởi sự tương hỗ giữa tri thức đang có với tri thức cần có để thực hiện tương tác có hiệu quả.

3.1.3.1. Tính dễ học (5)

- * *Tính khái quát*: Người dùng thường khái quát những tri thức và kinh nghiệm của mình từ một ứng dụng cụ thể vào những tình huống mới tương tự. Hệ thống có tính khái quát phải hỗ trợ hoạt động này.
- * *Tính nhất quán*:
 - Hệ thống nên hỗ trợ việc tạo nên kỹ năng cho người dùng bằng cách hoạt động tương tự trong những tình huống tương tự.
 - Tính nhất quán phải được áp dụng trong nhiều tình huống: cách đặt tên, cách cung cấp tham số cho lệnh, ...

3.1.3.2. Tính mềm dẻo (1)

- Liên quan đến nhiều cách mà người dùng và máy trao đổi thông tin.
- Thể hiện qua khả năng đáp ứng của hệ thống với những người dùng khác nhau trong những tình huống khác nhau.
- 5 nguyên tắc thể hiện tính mềm dẻo:
 - Đối thoại chủ động
 - Đa luồng
 - Khả năng chuyển giao giữa các nhiệm vụ
 - Khả năng thay thế
 - Khả năng tùy biến

3.1.3.2. Tính mềm dẻo (2)

* *Đối thoại chủ động:*

- *Đối thoại hướng hệ thống:*

- Hệ thống chủ động mọi đối thoại, người dùng đơn thuần đáp ứng các yêu cầu của hệ thống.

- Kiểu đối thoại này có tính mềm dẻo không cao.

- *Đối thoại hướng người dùng:*

- Người dùng hoàn toàn chủ động trong giao tiếp, cho phép người dùng cung cấp mọi hành động đầu vào ở bất kỳ thời điểm nào.

- Tăng tính mềm dẻo của hệ thống.

3.1.3.2. Tính mềm dẻo (3)

* Đa luồng:

- Luồng: là một phần của đối thoại, liên quan đến một nhiệm vụ đã cho của người dùng.
- Đa luồng: cho phép hỗ trợ nhiều hơn một nhiệm vụ tại một thời điểm.
 - Đa luồng tương tranh: cho phép việc trao đổi diễn ra cùng một lúc với nhiều nhiệm vụ khác nhau.
 - Đa luồng đan xen: cho phép các nhiệm vụ tạm thời gói lên nhau, với điều kiện tại một thời điểm chỉ có đối thoại với một nhiệm vụ. Hệ thống cửa sổ hỗ trợ một cách tự nhiên đối thoại đa luồng đan xen.

3.1.3.2. Tính mềm dẻo (4)

- * *Khả năng chuyển giao nhiệm vụ:* là khả năng chuyển quyền điều khiển thực hiện nhiệm vụ giữa người dùng và hệ thống.
Ví dụ: hệ thống có khả năng tự động kiểm tra lỗi chính tả của một văn bản.
- * *Khả năng thay thế:* cho phép các giá trị vào ra tương đương có thể thay thế cho nhau tùy ứng dụng.
Ví dụ: đơn vị lẻ của một văn bản có thể chọn là inche, cm, pt.
- * *Khả năng tùy biến:* là khả năng hệ thống tự động thay đổi để thích nghi với nhu cầu của người dùng.

3.1.3.3. Tính vững chắc (1)

- Tính vững chắc của một hệ thống là khả năng hỗ trợ việc thực hiện thành công và khẳng định đích của nhiệm vụ.
- Một số tiêu chí thể hiện tính vững chắc:
 - Tính quan sát
 - Tính khôi phục
 - Tính đáp ứng
 - Tính thích nghi nhiệm vụ

3.1.3.3. Tính vững chắc (2)

* Tính quan sát:

- Cho phép người dùng trực tiếp nhận biết được trạng thái của hệ thống.
- Thiết kế hệ thống phải đảm bảo không chỉ hành động mà các trạng thái của hệ thống cũng quan sát được.
- 5 tiêu chí của tính quan sát:
 - Quét tuần tự
 - Ngâm định
 - Đạt được
 - Ổn định
 - Nhìn thấy thao tác

3.1.3.3. Tính vững chắc (3)

* Tính khôi phục:

- Khi thao tác, người dùng có thể gây lỗi và muốn sửa lỗi.
- Tính khôi phục là khả năng đạt tới đích mong muốn sau khi nhận ra một số lỗi trong các tương tác trước.
- Các hệ thống nên được thiết kế sao cho các lỗi có thể được ngăn chặn và sửa được.

Ví dụ: khôi phục một văn bản sau khi thao tác lỗi nhờ nhấn phím “undo” trong hệ soạn thảo văn bản.

3.1.3.3. Tính vững chắc (4)

* *Tính đáp ứng:*

- Thể hiện tốc độ giao tiếp giữa người dùng và hệ thống.
- Thời gian đáp ứng: là khoảng thời gian cần thiết để biểu diễn thay đổi trạng thái.
- Thời gian đáp ứng càng ngắn càng tốt.
- Nếu đáp ứng không tức thì, hệ thống cần có các chỉ thị để người dùng hiểu là hệ thống đã nhận được yêu cầu và đang xử lý.

3.1.3.3. Tính vững chắc (5)

* *Tính thích nghi nhiệm vụ:*

- Hệ thống phải cho phép người dùng thực hiện được bất cứ nhiệm vụ mong muốn nào như đã được xác định trong bước phân tích nhiệm vụ.
- Đồng thời, hệ thống cũng phải cho phép người dùng có thể định nghĩa các nhiệm vụ mới liên quan đến lĩnh vực công việc đang thực hiện.

Chương 3 – THIẾT KẾ TƯƠNG TÁC NGƯỜI - MÁY

3.1. Các nguyên tắc khi thiết kế tương tác người - máy

3.2. Quy trình thiết kế phần mềm

3.3. Mô hình người dùng

3.4. Phân cấp mục tiêu và nhiệm vụ

3.5. Mô hình ngôn ngữ

3.6. Mô hình mức vật lý

Chương 3 – THIẾT KẾ TƯƠNG TÁC NGƯỜI - MÁY

3.2. Quy trình thiết kế phần mềm

3.2.1. Tổng quan về quy trình thiết kế

3.2.2. Vòng đời trong thiết kế

3.2.3. Thiết kế lập và mẫu thử

3.2.1. Tổng quan về quy trình thiết kế

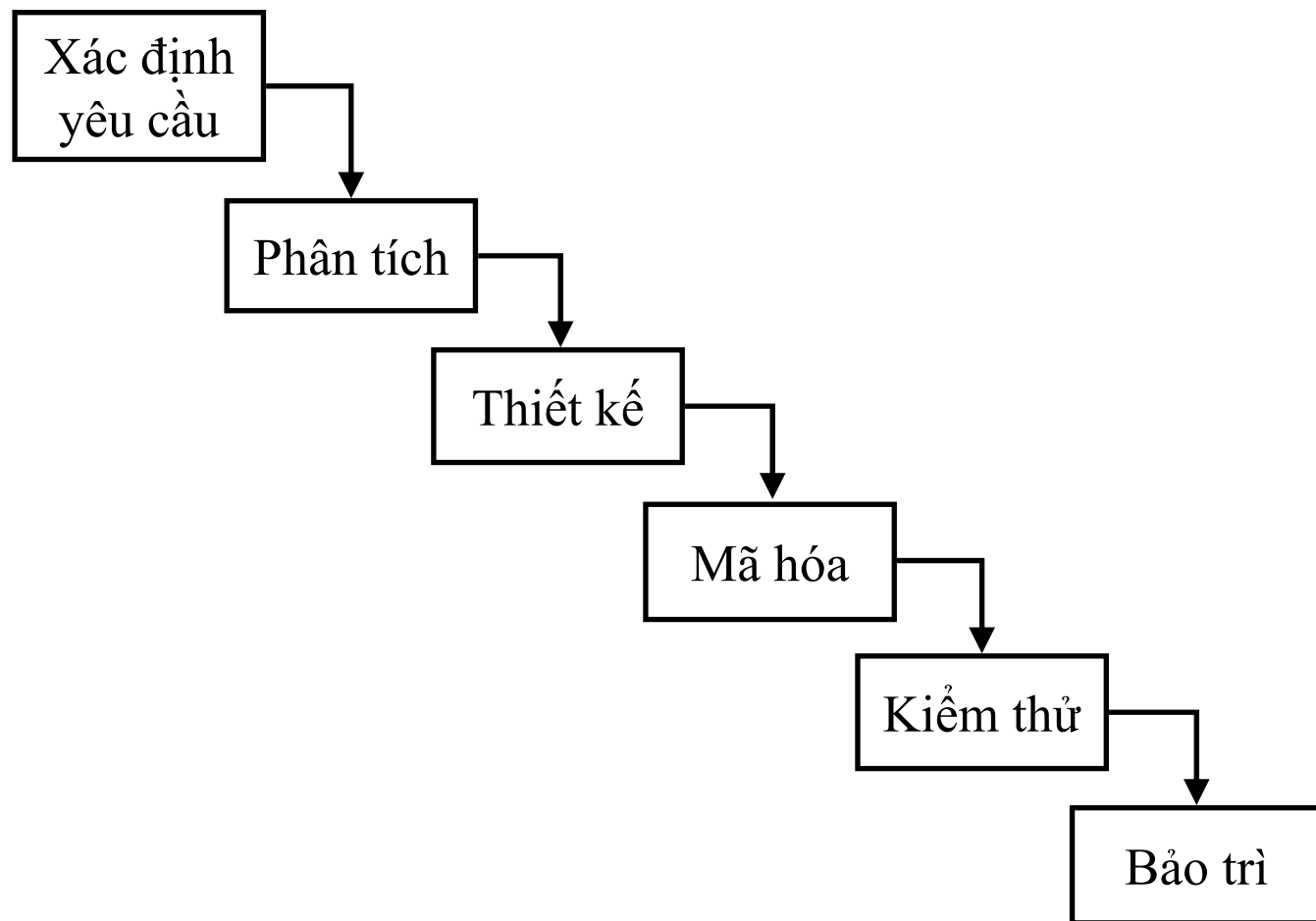
- Công nghệ phần mềm cung cấp phương tiện để hiểu cấu trúc của quy trình thiết kế.
- Quy trình thiết kế giúp khẳng định tính hiệu quả trong thiết kế HCI.
- Thiết kế liên quan đến quá trình phát triển một sản phẩm, một hệ thống. Các cách biểu diễn khác nhau của một hệ thống sẽ được tạo ra trong quá trình thiết kế.
- Một số khái niệm: vòng đời, mô hình,...
- Mục đích của thiết kế hệ thống tương tác: đảm bảo tính tiện dụng tối đa.

3.2.2. Vòng đời trong thiết kế

- Quan điểm chung nhất trong công nghệ phần mềm: quá trình phát triển hệ thống phần mềm bao gồm nhiều giai đoạn.
- Vòng đời phần mềm: là khoảng thời gian bắt đầu có yêu cầu xây dựng phần mềm đến khi có phần mềm, phần mềm được khai thác rồi chết đi.
- Một số mô hình vòng đời phần mềm:
 - Mô hình thác nước
 - Mô hình vòng đời phần mềm của Bohem
 - Mô hình vòng đời hình sao

3.2.2.1. Mô hình thác nước (1)

- Phát triển hệ thống phần mềm được tiến hành qua nhiều giai đoạn.
- Các giai đoạn là tuyến tính.



3.2.2.1. Mô hình thác nước (2)

- Xác định yêu cầu hệ thống:
 - Thiết lập yêu cầu cho mọi phần tử của hệ thống.
 - Cấp phát một tập con các yêu cầu đó cho phần mềm.
- Phân tích yêu cầu phần mềm:
 - Kỹ sư phần mềm phải hiểu về lĩnh vực thông tin đối với phần mềm, chức năng cần có, hiệu năng, giao diện.
 - Phải lập tư liệu về các yêu cầu cho cả hệ thống và phần mềm và đưa khách hàng duyệt.

3.2.2.1. Mô hình thác nước (3)

- Thiết kế phần mềm:
 - Tập trung vào 4 thuộc tính phân biệt của chương trình: cấu trúc dữ liệu, kiến trúc phần mềm, chi tiết thủ tục, đặc trưng giao diện.
 - Thiết kế là dịch các yêu cầu thành một biểu diễn của một phần mềm.
- Mã hóa:
 - Thực hiện nhiệm vụ dịch thiết kế thành dạng ngôn ngữ mà máy đọc được.

3.2.2.1. Mô hình thác nước (4)

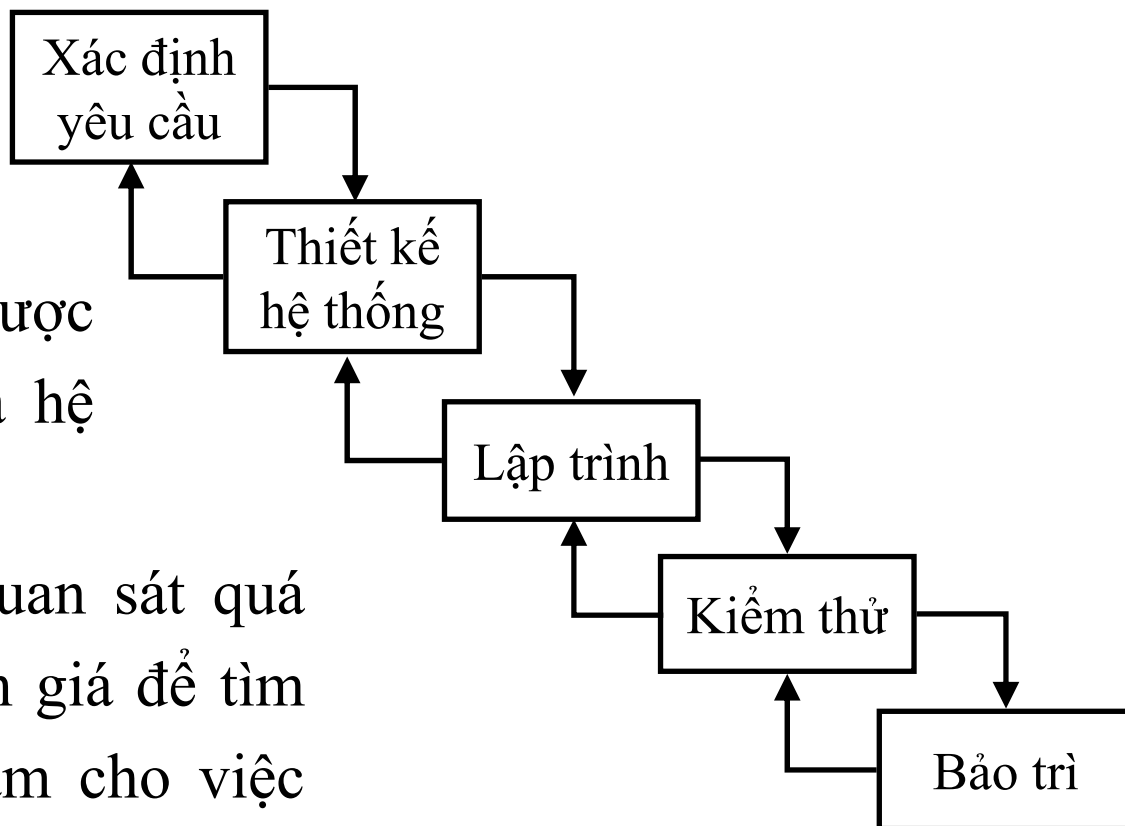
- Kiểm thử:
 - Tập trung vào phần logic bên trong của phần mềm.
 - Đảm bảo tất cả các câu lệnh đều được kiểm thử.
 - Về chức năng: đảm bảo phát hiện ra các lỗi (nếu có); đảm bảo với các đầu vào xác định, hệ thống cho kết quả thực tế giống với kết quả mong đợi.
- Bảo trì:
 - Áp dụng lại các bước vòng đời nêu trên cho chương trình hiện tại để đảm bảo hệ thống vẫn hoạt động tốt sau khi bàn giao cho khách hàng.

3.2.2.1. Mô hình thác nước (5)

- Mô hình thác nước là vòng đời cổ điển, và đã được sử dụng khá phổ biến.
- Một số vấn đề thường gặp khi sử dụng mô hình thác nước:
 - Ranh giới giữa các giai đoạn thường không rõ ràng, do đó khó thực hiện các giai đoạn một cách tuần tự.
 - Thường có phản hồi từ giai đoạn sau về giai đoạn trước.
 - Mô hình thác nước đòi hỏi khách hàng phát biểu mọi yêu cầu một cách tường minh. Điều này rất khó thực hiện ở giai đoạn đầu của dự án.
 - Chương trình chỉ hoàn thiện và hoạt động được ở giai đoạn cuối cùng của dự án.

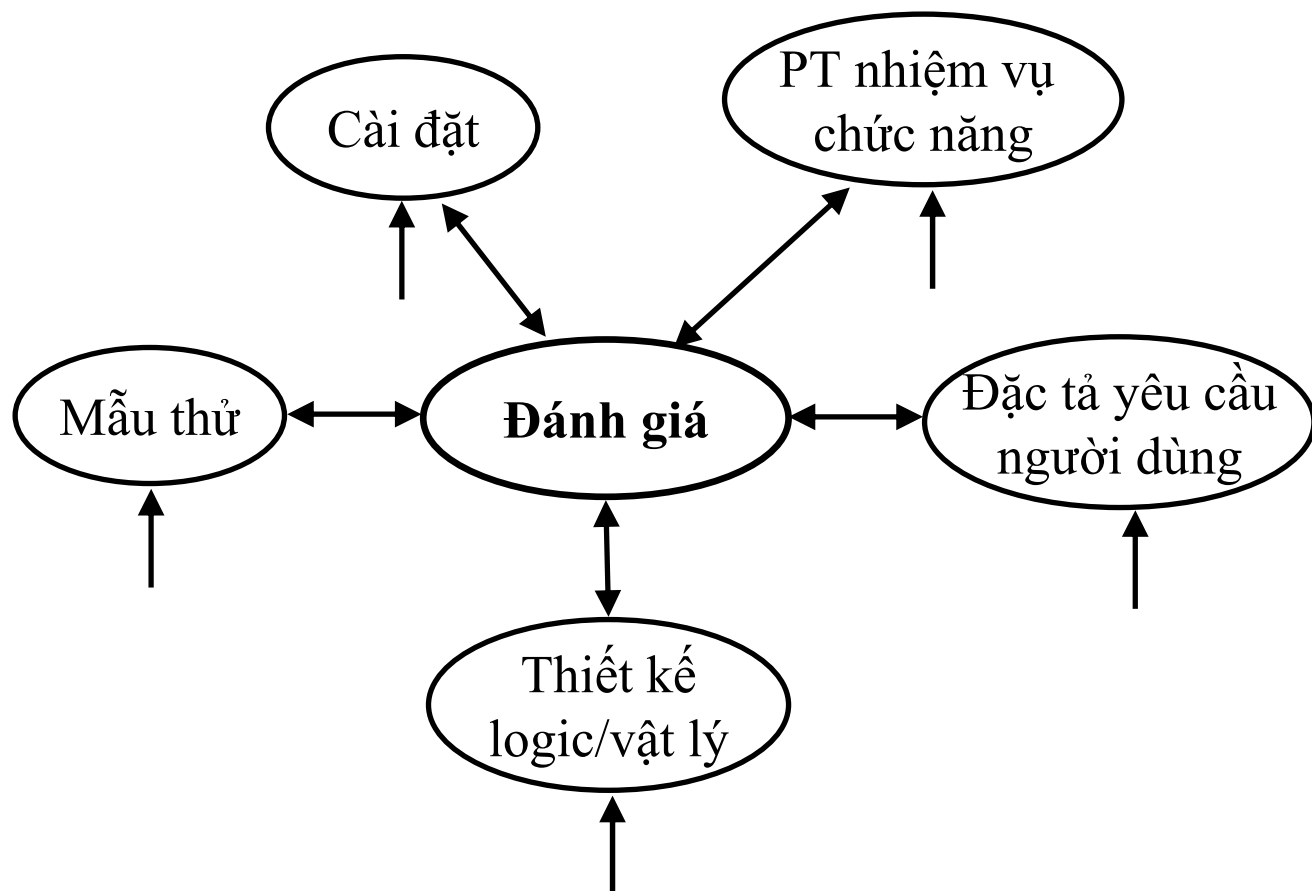
3.2.2.2. Mô hình vòng đời phần mềm của Bohem

- Có phản hồi từ giai đoạn sau về giai đoạn trước.
- Không thể xác định được tất cả các yêu cầu của hệ thống ngay từ đầu.
- Xây dựng hệ thống, quan sát quá trình tương tác và đánh giá để tìm ra các phương pháp làm cho việc tương tác được dễ dàng hơn.



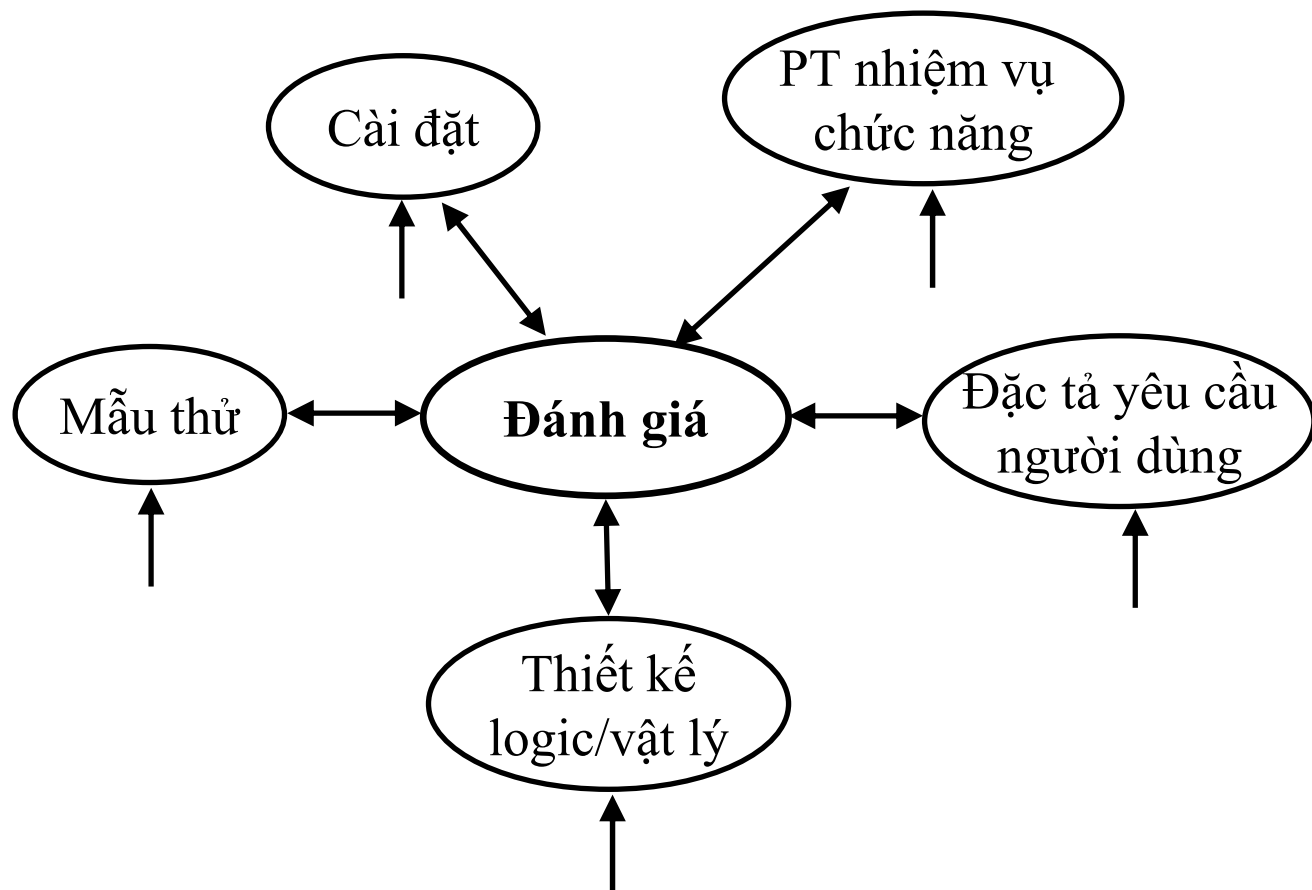
3.2.2.3. Mô hình vòng đời hình sao (1)

- Lấy người dùng làm trung tâm, coi người dùng là mục đích của thiết kế.
- Vòng đời hình sao do Hix & Hartson đề xuất năm 1993.



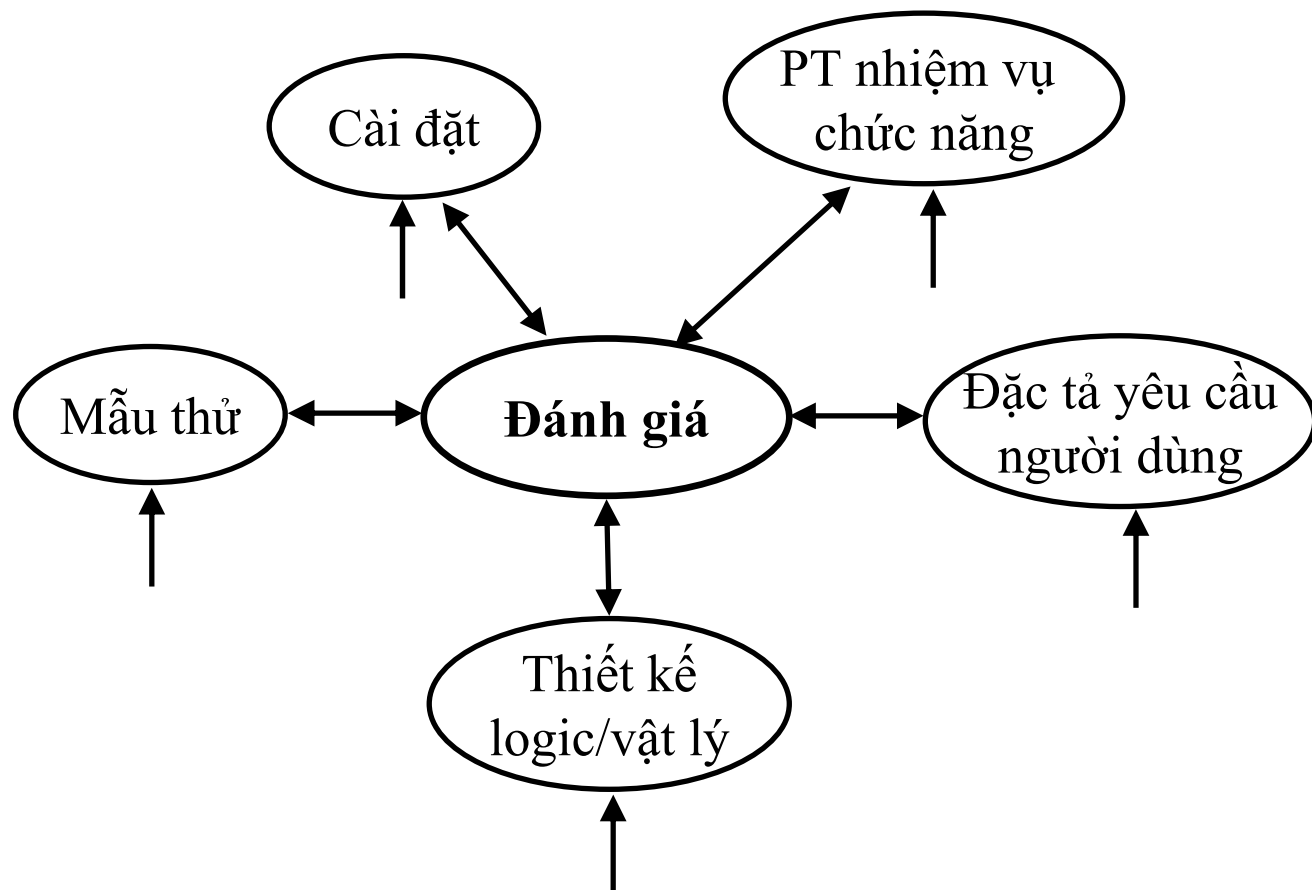
3.2.2.3. Mô hình vòng đời hình sao (2)

- Người dùng không chỉ bình luận về ý tưởng của người thiết kế, mà còn tham gia vào mọi khía cạnh của quá trình thiết kế.
- Thiết kế là thiết kế lặp.



3.2.2.3. Mô hình vòng đời hình sao (3)

- Thiết kế phải tích hợp được tri thức của người dùng và các chuyên gia từ nhiều lĩnh vực.
- Việc kiểm thử, đánh giá phải thực hiện ngay trong quá trình thiết kế.



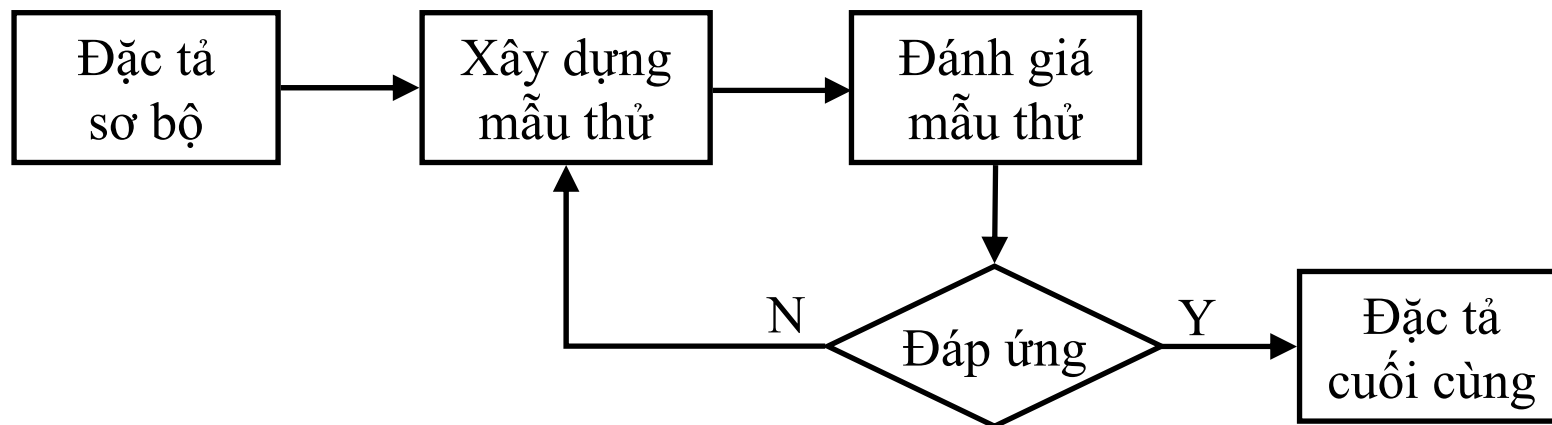
3.2.3. *Thiết kế lặp và mẫu thử (1)*

- Đối với các hệ tương tác, các yêu cầu người dùng khó có thể được xác định đầy đủ ngay từ đầu.
- Cách tốt nhất để đảm bảo các đặc trưng của thiết kế là xây dựng, thử nghiệm và đánh giá với người dùng thực sự. Sau đó hiệu chỉnh thiết kế để sửa các lỗi phát hiện được trong kiểm thử.
- Quá trình này được lặp đi lặp lại nhiều lần, gọi là thiết kế lặp.

3.2.3. Thiết kế lặp và mẫu thử (2)

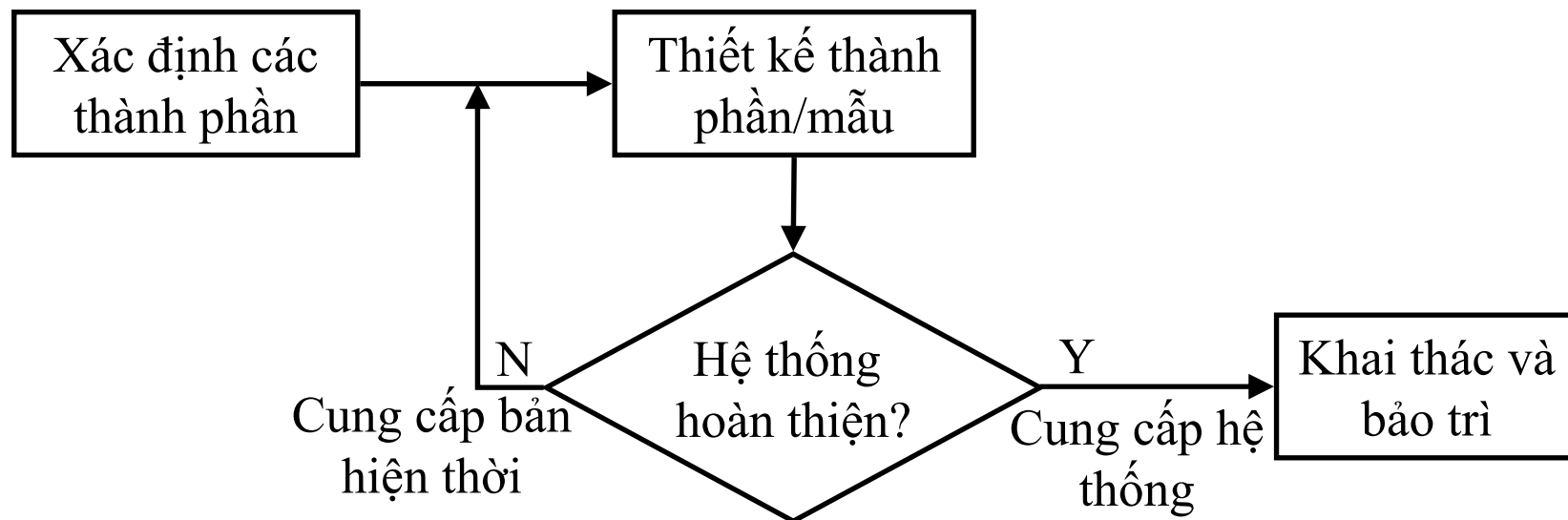
- Về mặt kỹ thuật, thiết kế lặp được thực hiện thông qua việc sử dụng các mẫu thử.
- Mẫu thử: là sự bắt chước hay mô phỏng một số chức năng đặc trưng của hệ thống (không phải là hệ thống đầy đủ).
- Ba kỹ thuật mẫu thử:
 - Mẫu thử kiểu hủy bỏ (throw away)
 - Mẫu thử kiểu gia tăng (incremental)
 - Mẫu thử kiểu tiến hóa (evolutionary)

3.2.3.1. Mẫu thử kiểu hủy bỏ (throw away)



- Mẫu thử được xây dựng và thử nghiệm.
- Kiến thức thu được từ quá trình này được sử dụng để xây dựng sản phẩm cuối cùng.
- Mẫu thử sẽ bị hủy bỏ.

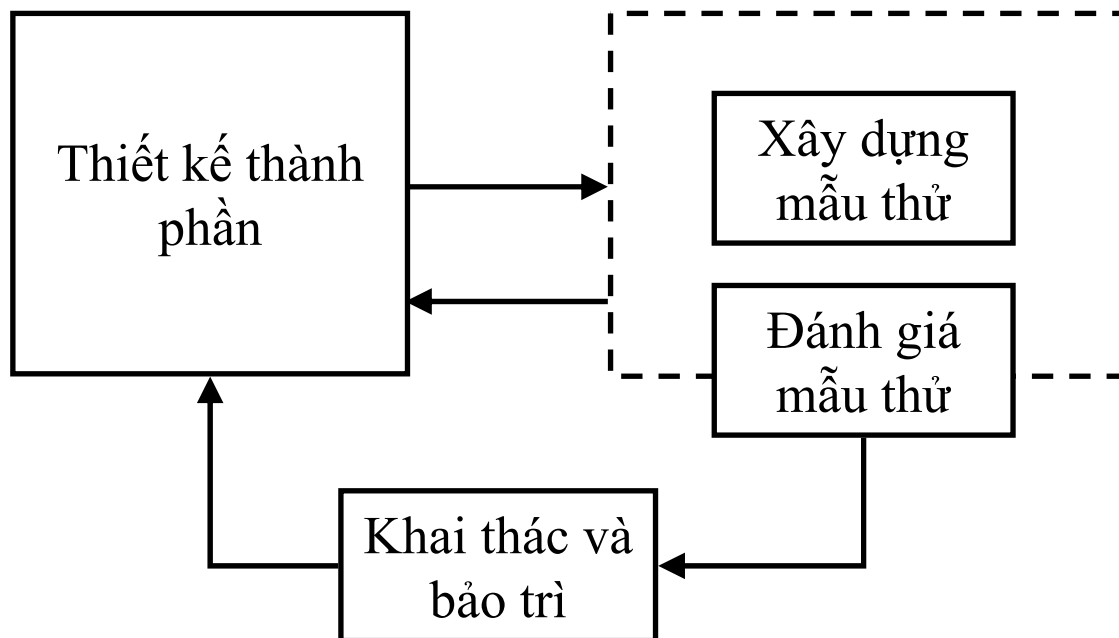
3.2.3.2. Mẫu thử kiểu gia tăng (incremental)



- Thiết kế được chia thành các thành phần nhỏ và độc lập.
- Sản phẩm cuối cùng là một chuỗi sản phẩm, mỗi sản phẩm có thêm một thành phần.

3.2.3.3. Mẫu thử kiểu tiến hóa (evolutionary)

- Mẫu thử không bị hủy bỏ mà được dùng làm cơ sở cho lần lặp tiếp theo.
- Hệ thống hiện thời là sự tiến hóa từ phiên bản ban đầu đến sản phẩm cuối cùng.



Ưu điểm của việc sử dụng mẫu thử:

- Mẫu thử khác sản phẩm cuối cùng ở một số chức năng cũng như tính hiệu quả.
- Một số ưu điểm khi sử dụng mẫu thử:
 - Làm mịn đặc tả.
 - Làm mịn thiết kế.
 - So sánh, đánh giá các thiết kế.
 - Chứng minh cho một ý tưởng.
 - Sử dụng cho các đối tượng: người dùng, nhà thiết kế,...

Nhược điểm của việc sử dụng mẫu thử:

- Tốn thời gian, đặc biệt với mẫu thử kiểu hủy bỏ.
- Yêu cầu người quản lý dự án phải có kinh nghiệm cần thiết để lập kế hoạch phù hợp và dự toán chi phí cho quá trình thiết kế.
- Một số phần quan trọng của hệ thống lại không phải là các tính năng (ví dụ: độ an toàn, độ tin cậy). Sử dụng mẫu thử không kiểm tra được các thành phần phi chức năng này.

Chương 3 – THIẾT KẾ TƯƠNG TÁC NGƯỜI - MÁY

3.1. Các nguyên tắc khi thiết kế tương tác người - máy

3.2. Quy trình thiết kế

3.3. Mô hình người dùng

3.4. Phân cấp mục tiêu và nhiệm vụ

3.5. Mô hình ngôn ngữ

3.6. Mô hình mức vật lý

Chương 3 – THIẾT KẾ TƯƠNG TÁC NGƯỜI - MÁY

3.3. Mô hình người dùng

3.3.1. *Tổng quan*

3.3.2. *Mô hình đặc tả yêu cầu người dùng*

3.3.3. *Mô hình nhận thức*

3.3.1. Tổng quan (1)

- Người thiết kế luôn lựa chọn và sử dụng các mô hình thích hợp trong quá trình thiết kế.
- Một số mô hình mang tính đánh giá, nghĩa là cho biết một thiết kế có chính xác hay không.
- Một số mô hình mang tính khởi sinh, nghĩa là có đóng góp cho quá trình thiết kế.

3.3.1. Tổng quan (2)

- Hai nhóm mô hình trong giao tiếp người dùng – máy tính:
- Mô hình đặc tả yêu cầu người dùng: dùng để thiết lập nhu cầu người dùng.
- Mô hình nhận thức: biểu diễn người dùng trong một hệ thống tương tác, nhằm hiểu được các sắc thái người dùng trong nhận thức, trong tri thức và trong xử lý.

3.3.2. Mô hình đặc tả yêu cầu người dùng (1)

- Xác định yêu cầu người dùng là một nhiệm vụ quan trọng trong việc xây dựng một phần mềm.
- Quá trình này chỉ tập trung xác định các yêu cầu chức năng (nhiệm vụ của hệ thống), ít khi chú ý đến các yêu cầu phi chức năng của hệ thống (như tính tiện dụng, độ an toàn).
- Mô hình đặc tả yêu cầu người dùng: coi người dùng là một thành viên của quá trình thiết kế; người dùng không chỉ bình luận, xem xét ý tưởng mà còn tham gia vào mọi khía cạnh của quá trình thiết kế.

3.3.2. Mô hình đặc tả yêu cầu người dùng (2)

- Một số mô hình đặc tả yêu cầu người dùng:
 - Mô hình kỹ thuật xã hội
 - Mô hình hệ thống phần mềm
 - Mô hình đa cách nhìn
 - Mô hình thiết kế hợp tác

3.3.2.1. Mô hình kỹ thuật xã hội (1)

- Quan tâm đến các khía cạnh của thiết kế về mặt kỹ thuật, xã hội, tổ chức và con người.
- Dựa trên quan điểm: công nghệ không phát triển một cách độc lập, riêng biệt, mà được phát triển trong một môi trường xã hội rộng rãi.
- Các mô hình kỹ thuật xã hội tiêu biểu:
 - OSTA (Open System Task Analysis)
 - USTM (User Skills and Task Match)
 - ETHICS (Effective Technical and Human Implementation of Computer Systems)

3.3.2.1. Mô hình kỹ thuật xã hội (2) - OSTA

- Mô hình OSTA (Open System Task Analysis) – Phân tích nhiệm vụ các hệ thống mở.
- Mô hình này do Eason và Harker đề xuất năm 1998 – 1999.
- OSTA thử mô tả điều gì sẽ xảy ra khi một hệ thống kỹ thuật được đưa vào môi trường kỹ thuật của một tổ chức.
- Trong OSTA, các khía cạnh xã hội của hệ thống (như tính tiện dụng, độ an toàn) được xác định cùng với các yêu cầu kỹ thuật (cấu trúc, chức năng hệ thống).

3.3.2.1. Mô hình kỹ thuật xã hội (3) - OSTA

- OSTA gồm 8 giai đoạn chính:
 - Giai đoạn 1: Xác định nhiệm vụ chính mà công nghệ phải hỗ trợ theo mục đích của người sử dụng.
 - Giai đoạn 2: Xác định đầu vào của hệ thống. Đầu vào này có thể ở nhiều dạng khác nhau, phụ thuộc vào thiết kế.
 - Giai đoạn 3: Mô tả môi trường bên ngoài của tổ chức nơi mà hệ thống sẽ được đưa vào áp dụng, bao gồm các khía cạnh: tự nhiên, kinh tế và chính trị.
 - Giai đoạn 4: Mô tả quá trình chuyển động bên trong hệ thống (mô tả chức năng), thường được biểu diễn dưới dạng lưu đồ.

3.3.2.1. Mô hình kỹ thuật xã hội (3) - OSTA

- OSTA gồm 8 giai đoạn chính (tiếp):
 - Giai đoạn 5: Phân tích hệ thống xã hội, có xem xét đến các nhóm làm việc hiện tại và các mối quan hệ bên trong và bên ngoài tổ chức.
 - Giai đoạn 6: Mô tả về cấu hình và sự tương tác của hệ thống kỹ thuật với các hệ thống khác.
 - Giai đoạn 7: Thiết lập các tiêu chuẩn đánh giá sự hài lòng về hệ thống, chỉ ra các yêu cầu về xã hội và kỹ thuật của hệ thống.
 - Giai đoạn 8: Chỉ ra các yêu cầu của hệ thống kỹ thuật mới.

3.3.2.1. Mô hình kỹ thuật xã hội (4) – USTM/CUSTOM

- Mô hình USTM (User Skills and Task Match) – Các kỹ năng người dùng và sự phù hợp nhiệm vụ.
- Đây là mô hình thiết kế giúp cho đội thiết kế hiểu và cung cấp tài liệu một cách đầy đủ về các yêu cầu của người dùng.
- Sử dụng các mô hình nhiệm vụ dưới dạng biểu đồ và các mô tả chi tiết.
- USTM được biến đổi thành CUSTOM cho những tổ chức nhỏ hơn.

3.3.2.1. Mô hình kỹ thuật xã hội (5) – USTM/CUSTOM

- Mô hình USTM/CUSTOM tập trung vào tất cả những “người liên quan”, chứ không chỉ tập trung vào người dùng.
- “Người liên quan”: là bất cứ người nào liên quan đến sự thành công hay thất bại của phần mềm.
- Yêu cầu của tất cả người liên quan đều được xét đến, không chỉ yêu cầu của người dùng cuối.
- Người liên quan được chia làm 4 nhóm.

3.3.2.1. Mô hình kỹ thuật xã hội (6) – USTM/CUSTOM

- 4 nhóm người liên quan:
 - Nhóm 1: những người sử dụng hệ thống.
 - Nhóm 2: những người không trực tiếp sử dụng hệ thống, nhưng có nhận thông tin từ đầu ra hệ thống, hoặc cung cấp thông tin đầu vào cho hệ thống.
 - Nhóm 3: không thuộc hai nhóm trên, nhưng chịu tác động từ sự thành công hay thất bại của hệ thống.
 - Nhóm 4: những người tham gia vào quá trình thiết kế, phát triển và bảo trì hệ thống.

3.3.2.1. Mô hình kỹ thuật xã hội (7) – USTM/CUSTOM

- CUSTOM được áp dụng vào giai đoạn đầu của quá trình thiết kế.
- Mô hình này dựa trên các định dạng, cung cấp một tập các câu hỏi:
 1. Mô tả ngữ cảnh về tổ chức, bao gồm những mục tiêu chủ yếu, đặc trưng vật lý, nền móng chính trị và kinh tế.
 2. Xác định và mô tả người liên quan. Tất cả những người liên quan đều được đặt tên, phân thành các nhóm (nhóm 1, nhóm 2, nhóm 3, nhóm 4). Mô tả các vấn đề cá nhân, chức năng, nhiệm vụ của họ trong tổ chức.

3.3.2.1. Mô hình kỹ thuật xã hội (8) – USTM/CUSTOM

- Tập câu hỏi của mô hình CUSTOM (tiếp):

3. Xác định và mô tả các nhóm làm việc. Một nhóm làm việc là một nhóm người bất kỳ làm chung một nhiệm vụ, cho dù họ có được thành lập chính thức hay không. Các nhóm làm việc được mô tả theo vai trò của họ trong tổ chức và đặc điểm của họ.

4. Xác định và mô tả các cặp nhiệm vụ - đối tượng. Đây là những nhiệm vụ phải thực hiện, liên kết với những đối tượng được dùng để thực hiện nhiệm vụ; hoặc với những đối tượng mà nhiệm vụ được áp dụng vào.

3.3.2.1. Mô hình kỹ thuật xã hội (9) – USTM/CUSTOM

- Tập câu hỏi của mô hình CUSTOM (tiếp):

5. Xác định nhu cầu của “người liên quan”. Các bước từ 2 đến 4 được mô tả dựa trên hệ thống hiện tại và hệ thống đang muốn xây dựng. Nhu cầu của “người liên quan” được xác định dựa trên sự khác nhau giữa hệ thống hiện tại và hệ thống đang xây dựng.

Ví dụ: hiện tại nếu có một người thiếu một kỹ năng được yêu cầu trong hệ thống mới thì việc đào tạo được xác định là một nhu cầu.

6. Tập hợp và kiểm tra các yêu cầu của “người liên quan”.

3.3.2.1. Mô hình kỹ thuật xã hội (10) – ETHICS

- ETHICS (Effective Technical and Human Implementation of Computer Systems) – Mô hình nâng cao hiệu quả thực hiện về kỹ thuật và con người của các hệ thống máy tính.
- Giống OSTA, ETHICS cũng liên quan đến việc xác định các yêu cầu kỹ thuật và xã hội.
- Khác OSTA, ETHICS sử dụng hai đội thiết kế làm việc độc lập với nhau để xác định những yêu cầu kỹ thuật và xã hội của thiết kế một cách song song.
- Kết quả làm việc của hai nhóm được kết hợp để tìm ra một giải pháp hiệu quả nhất phù hợp với những yêu cầu kỹ thuật – xã hội đã được xác định.

3.3.2.1. Mô hình kỹ thuật xã hội (11) – ETHICS

- 6 giai đoạn của ETHICS:

1. Xác định vấn đề và mô tả hệ thống hiện tại. Xác định mục tiêu và nhiệm vụ. Xác định những yếu tố ràng buộc của hệ thống, cả về kỹ thuật và xã hội.

2. Thành lập hai đội thiết kế. Một đội khảo sát những khía cạnh kỹ thuật, một đội khảo sát những khía cạnh xã hội. Các mục tiêu và nhu cầu xác định được trong bước 1 được sắp xếp theo thứ tự ưu tiên và được kiểm tra về tính tương thích trước khi đưa ra những quyết định thiết kế về xã hội và kỹ thuật.

3.3.2.1. Mô hình kỹ thuật xã hội (12) – ETHICS

- 6 giai đoạn của ETHICS (tiếp):

3. Đưa ra các giải pháp kỹ thuật và xã hội khác nhau, đánh giá các giải pháp này theo những tiêu chuẩn đã được thiết lập để lựa chọn danh sách rút gọn các giải pháp có khả năng.

4. Kiểm tra tính tương thích của các giải pháp được lựa chọn ở bước 3.

5. Dựa vào các tiêu chuẩn đánh giá, sắp xếp các cặp tương thích các giải pháp xã hội – kỹ thuật.

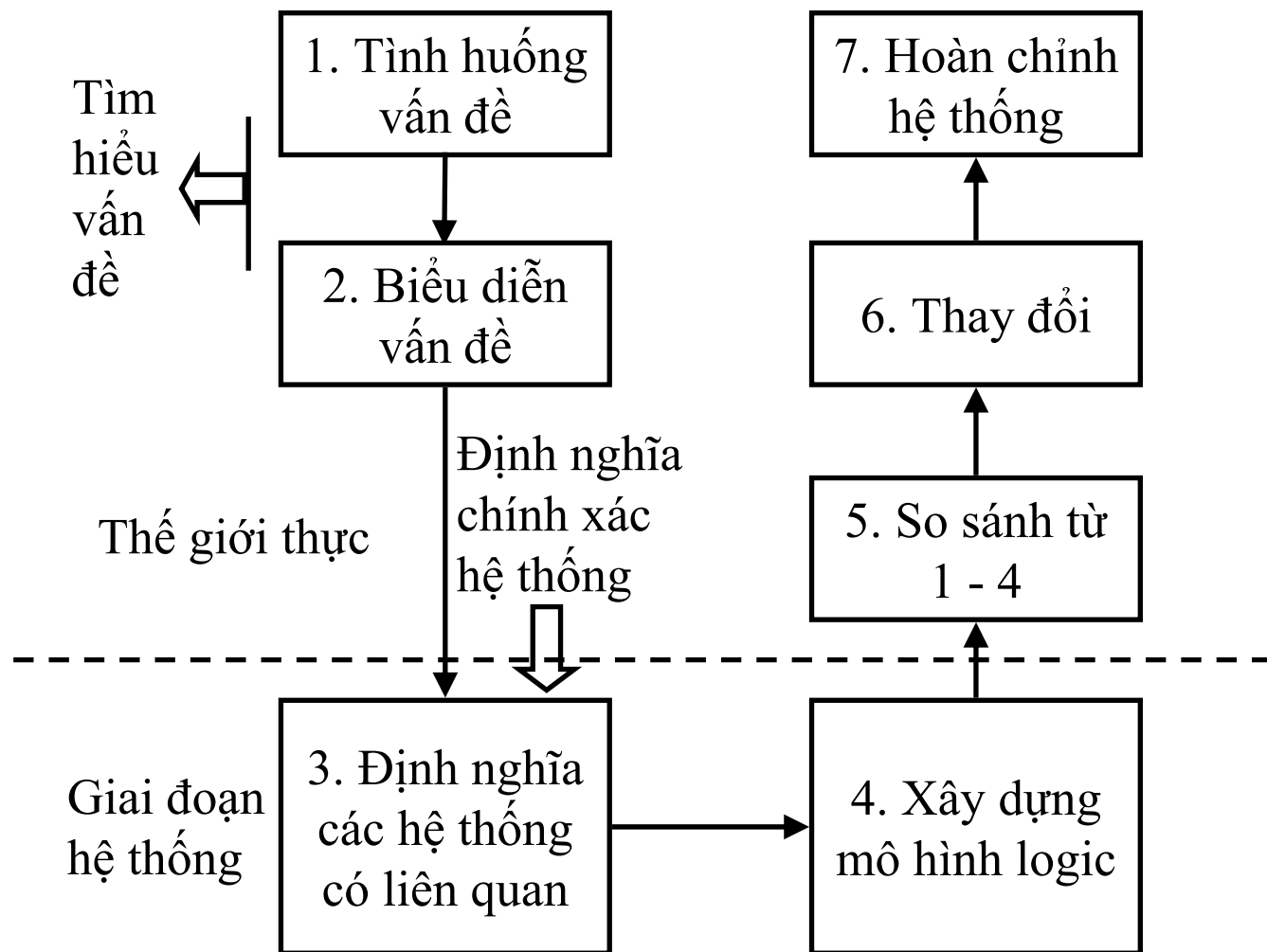
6. Phát triển thiết kế chi tiết.

3.3.2.2. Mô hình hệ thống phần mềm (1)

- Mô hình hệ thống phần mềm SSM (Soft System Methodology).
- SSM được đề xuất bởi Checkland và Scholes năm 1991 và được Eason phát triển vào năm 1992.
- SSM không chỉ tập trung xác định yêu cầu của người dùng cả về kỹ thuật và xã hội (như các mô hình kỹ thuật xã hội), mà xem xét một cách rộng hơn: coi tổ chức như một hệ thống, trong đó công nghệ và con người là các thành phần chính.

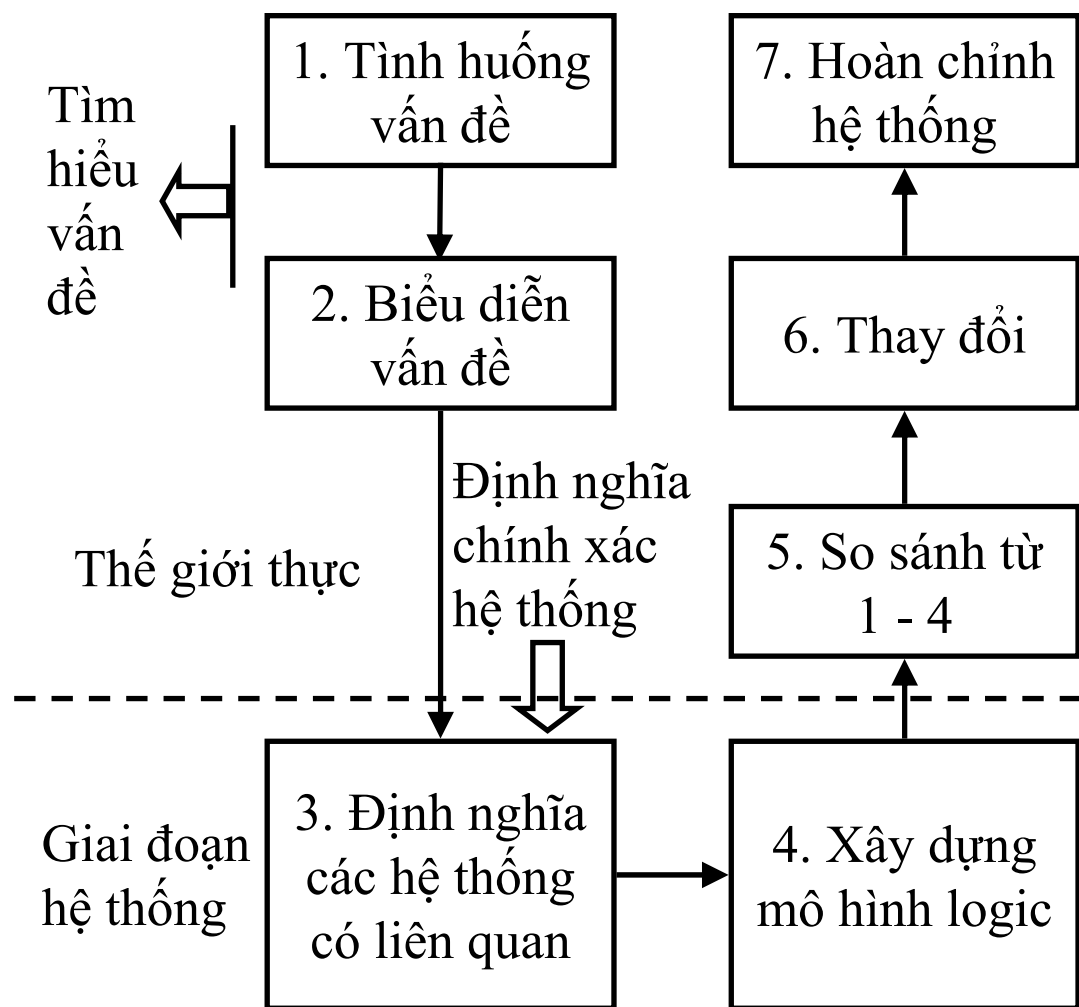
3.3.2.2. Mô hình hệ thống phần mềm (2)

- SSM gồm 7 bước chính.
- Giai đoạn thể giới thực: các bước 1-2; 5-7.
- Giai đoạn hệ thống (trừu tượng): các bước 3-4.



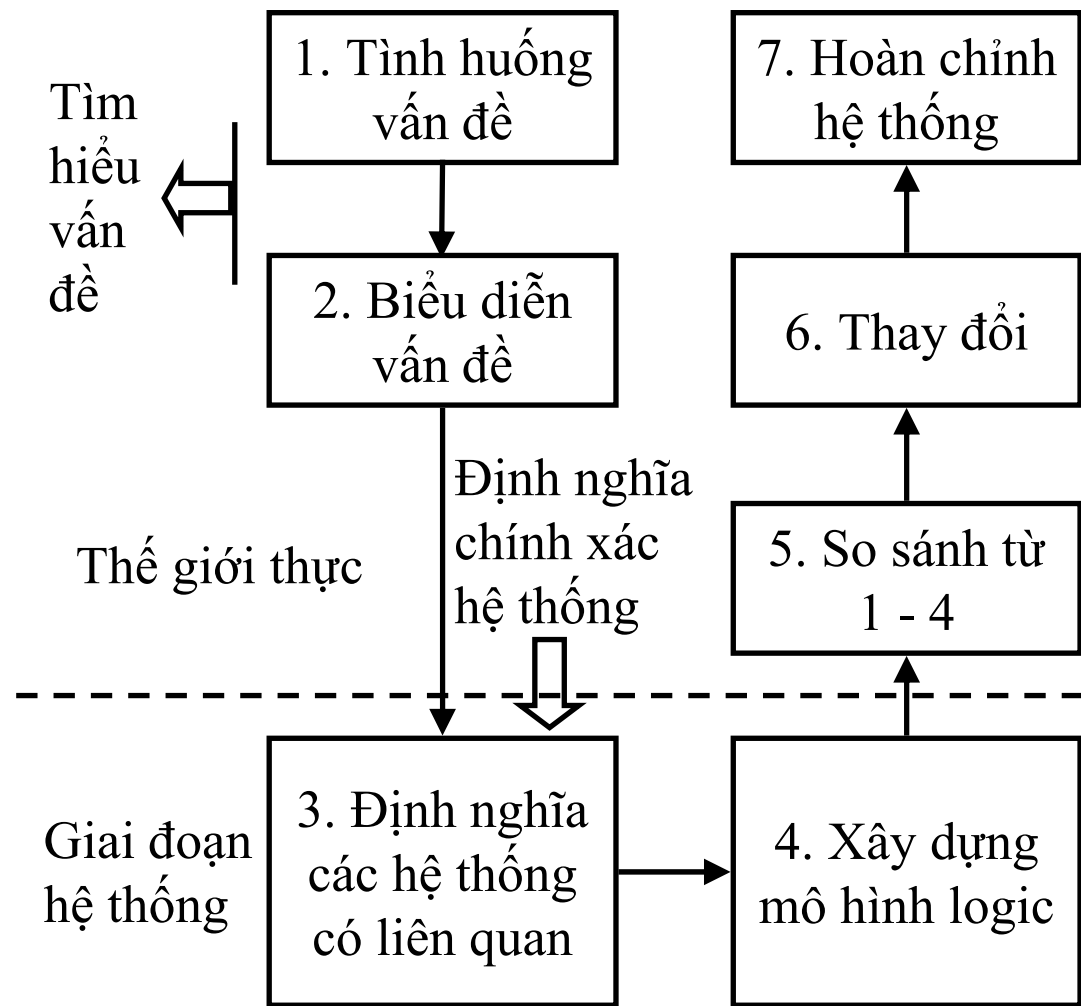
3.3.2.2. Mô hình hệ thống phần mềm (3)

- Bước 1: Xác định vấn đề.
- Bước 2: Phân tích vấn đề: Mô tả chi tiết về tình huống vấn đề.
 - Những người liên quan và nhiệm vụ của họ.
 - Các nhóm làm việc.
 - Cấu trúc tổ chức.
 - Các quy trình của tổ chức.
 - Các vấn đề được đưa ra bởi người liên quan.



3.3.2.2. Mô hình hệ thống phần mềm (4)

- Bước 3: Khởi tạo định nghĩa gốc cho hệ thống (chuyển từ thế giới thực sang giai đoạn hệ thống). Định nghĩa gốc được mô tả dưới dạng CATWOE.



3.3.2.2. Mô hình hệ thống phần mềm (5)

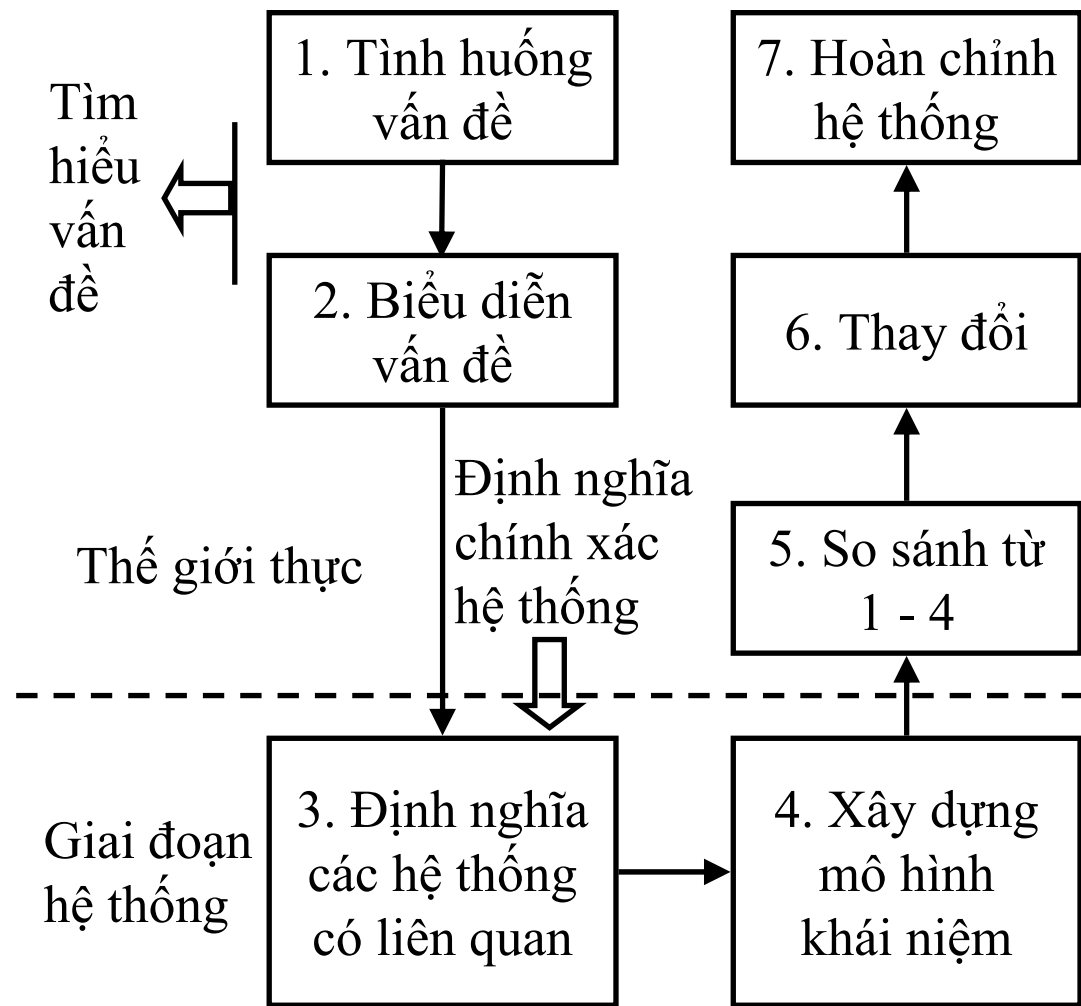
- Bước 3 (tiếp): CATWOE
 - **Clients** (khách hàng): là những người nhận thông tin đầu ra hay thu được lợi ích từ hệ thống.
 - **Actors** (người thực hiện): là những người thực hiện các hoạt động bên trong hệ thống.
 - **Transformations** (những biến đổi): là những thay đổi do tác động của hệ thống. Đây là một trong những định nghĩa gốc quan trọng, nó dẫn đến những hoạt động cần thiết cho các bước tiếp theo. Để xác định được những biến đổi, phải xem xét các đầu vào và đầu ra của hệ thống.

3.3.2.2. Mô hình hệ thống phần mềm (6)

- Bước 3 (tiếp): CATWOE
 - **Weltanschauung** (theo tiếng Đức) hoặc **World View** – quang cảnh thế giới: là cách cảm nhận hệ thống xét trong một định nghĩa gốc cụ thể nào đó.
 - **Owner** (người sở hữu): là những người sở hữu hoặc chịu trách nhiệm về hệ thống, hoặc là những người có quyền tạo ra những thay đổi đối với hệ thống.
 - **Environment** (môi trường): là môi trường mà hệ thống hoạt động và chịu ảnh hưởng.

3.3.2.2. Mô hình hệ thống phần mềm (7)

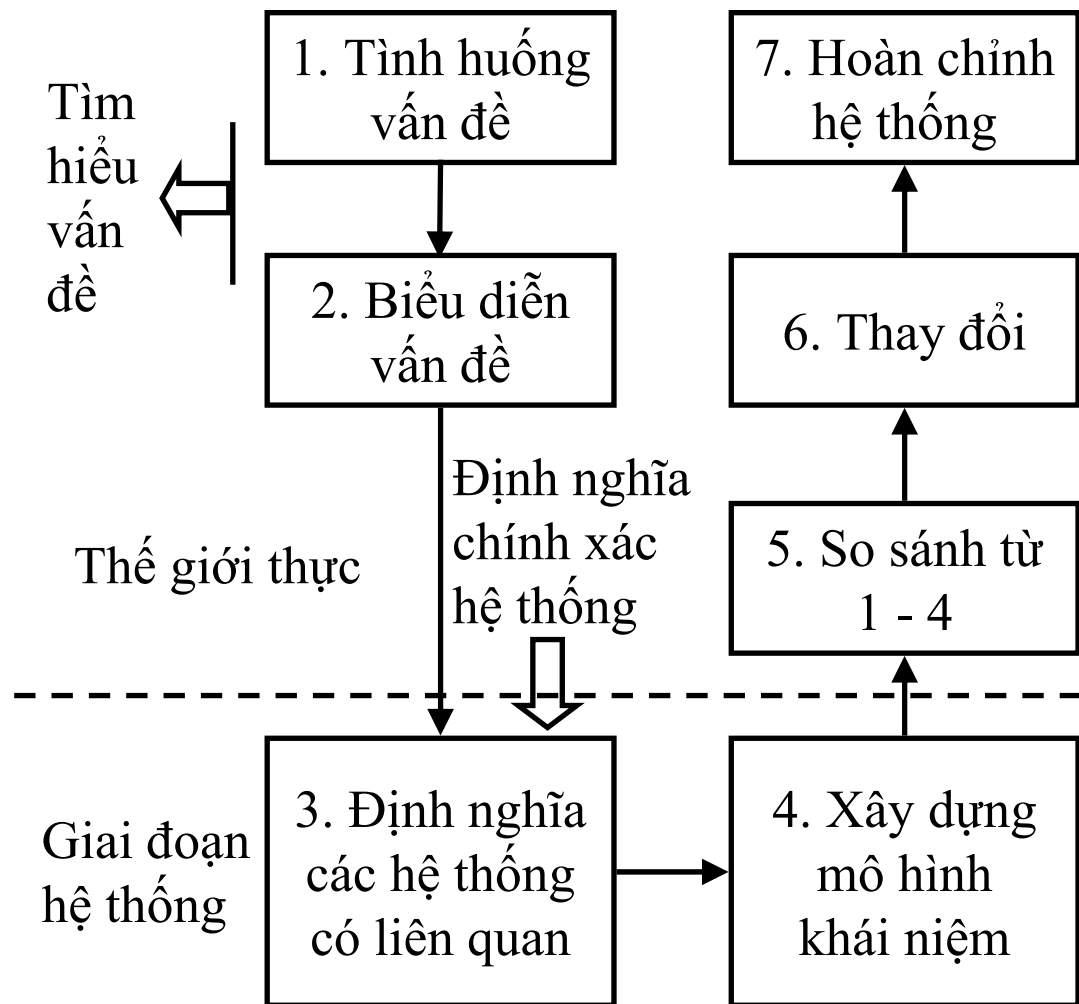
- Bước 4: Xây dựng mô hình khái niệm.
 - Mô hình khái niệm định nghĩa những việc mà hệ thống phải thực hiện để thỏa mãn các định nghĩa gốc.
 - Nhận dạng các thay đổi và các hoạt động trong hệ thống.
 - Mô hình hóa những gì hệ thống đạt được và cách thức thực hiện của hệ thống.



3.3.2.2. Mô hình hệ thống phần mềm (8)

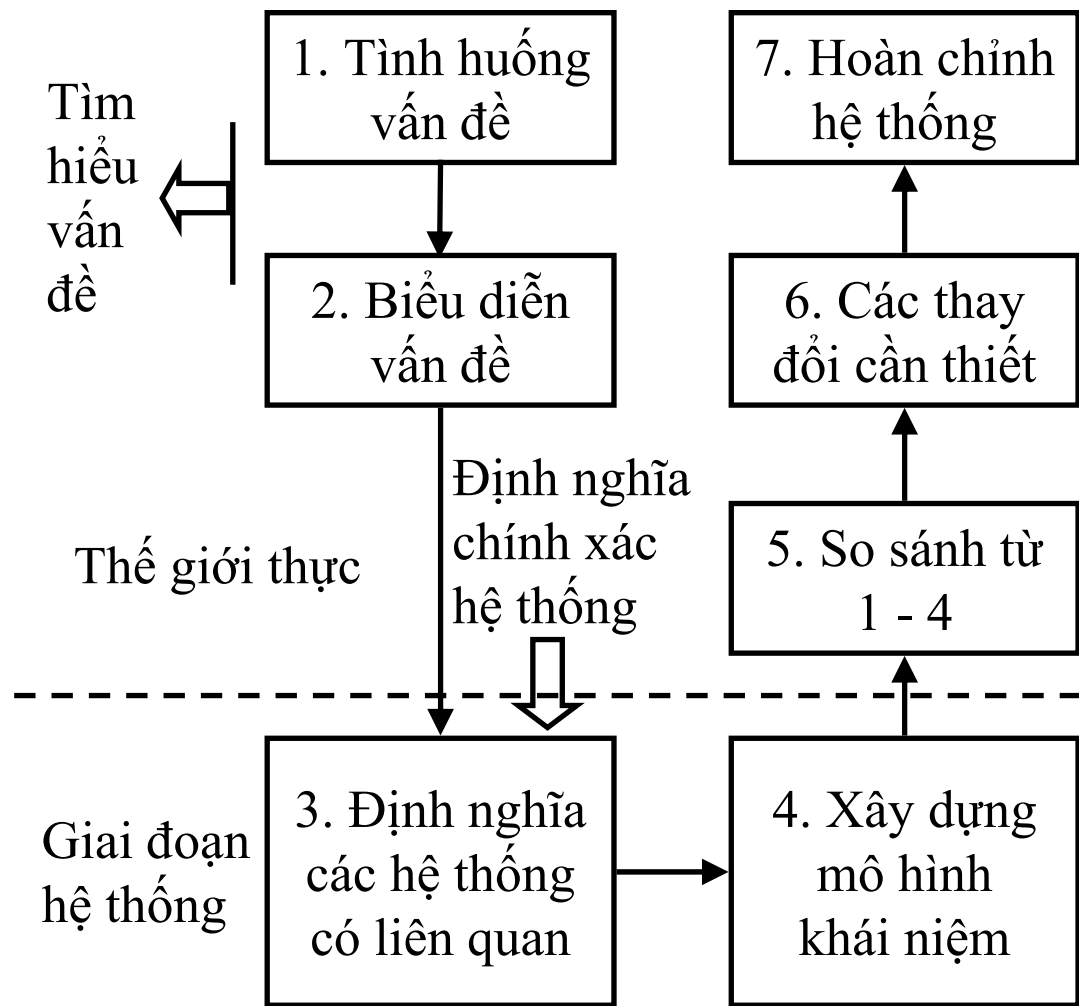
• Bước 5:

- So sánh hệ thống thực với hệ thống của mô hình khái niệm.
- Xác định sự khác nhau giữa hai hệ thống.



3.3.2.2. Mô hình hệ thống phần mềm (9)

- Bước 6: Xác định những thay đổi cần thiết và có ích cho hệ thống (về cấu trúc, thủ tục,...).
- Bước 7: Quyết định những hành động cần thiết để đạt được những thay đổi ở bước 6.



3.3.2.2. Mô hình hệ thống phần mềm (10)

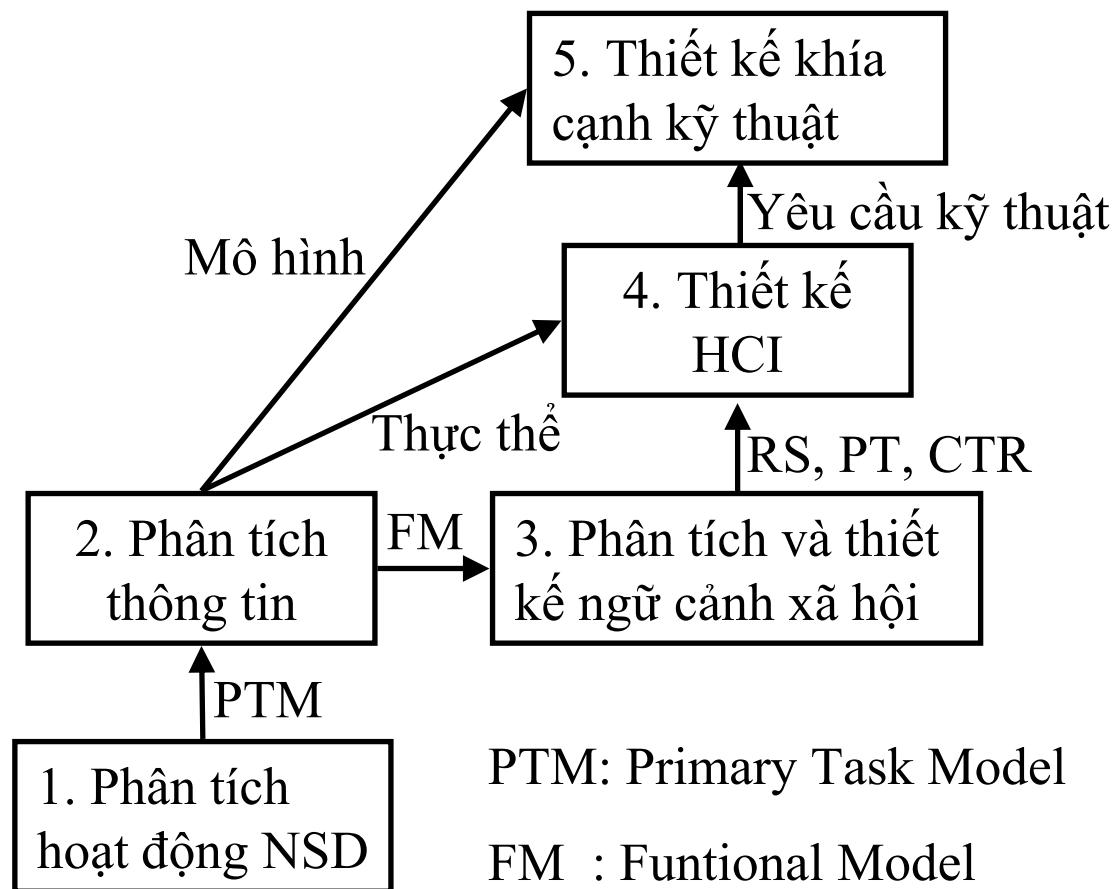
- SSM là cách tiếp cận mềm dẻo, linh hoạt.
- Hỗ trợ việc xem xét một cách chi tiết ngữ cảnh của quá trình thiết kế.
- SSM nhằm giúp cho người thiết kế hiểu rõ hơn về hệ thống mà họ đang phát triển.

3.3.2.3. Mô hình đa cách nhìn - Multiview (1)

- Mô hình đa cách nhìn: gắn hệ thống phần mềm và xã hội vào một phương pháp có tính hoàn thiện.
- Do Avison và Wood – Happer đưa ra năm 1990.
- Là một cách tiếp cận tổ hợp nhiều cách tiếp cận trong một giai đoạn, có phương pháp kiểm tra.
- Phương pháp này bao gồm cả cách tiếp cận kỹ thuật xã hội và các mô hình thực thể để biểu diễn cấu trúc thông tin.

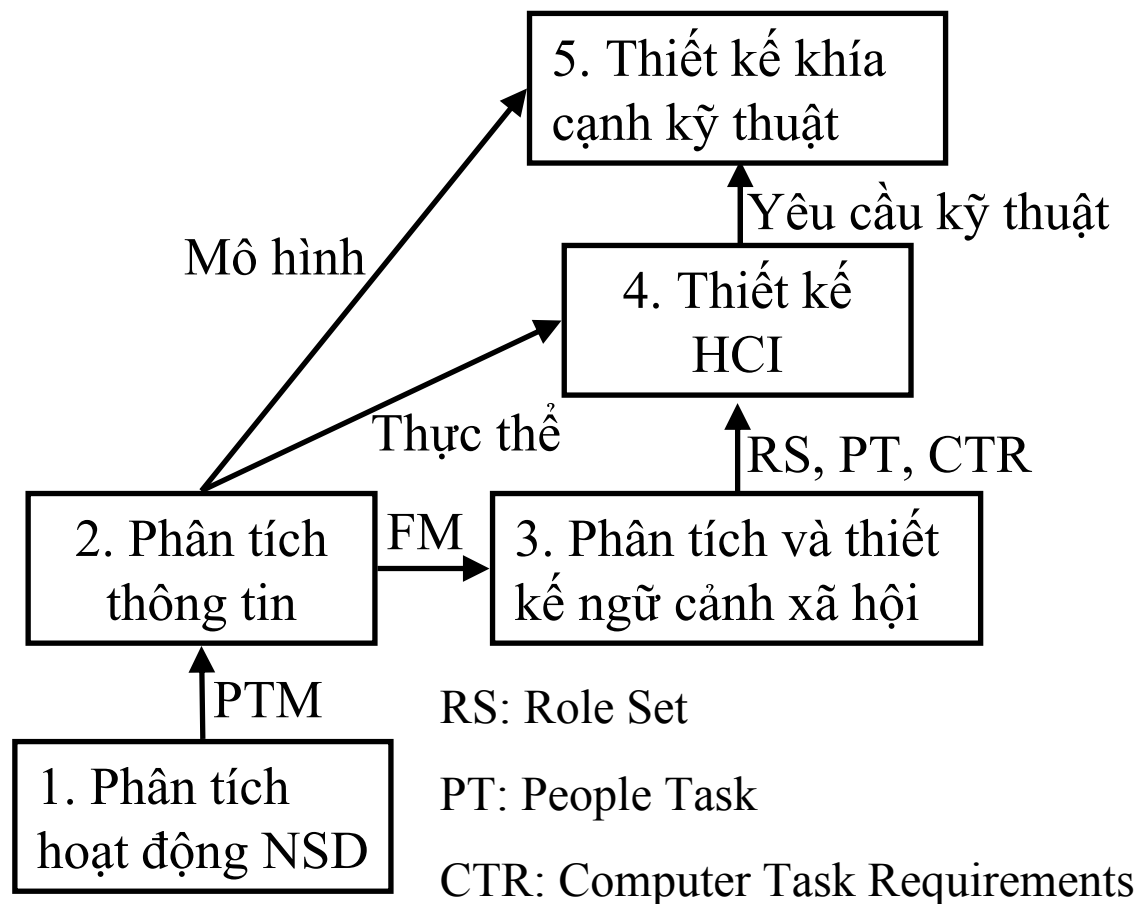
3.3.2.3. Mô hình đa cách nhìn - Multiview (2)

- Bước 1: Mô tả mục đích của hệ thống, bao gồm người liên quan và chủ hệ thống.
- Bước 2: Phân tích thông tin liên quan đến mô hình khái niệm của luồng dữ liệu và cấu trúc thông tin.



3.3.2.3. Mô hình đa cách nhìn - Multiview (3)

- Bước 3: Sử dụng mô hình chức năng để làm cơ sở cho phân phối nhiệm vụ người dùng.
- Bước 4: Thiết kế giao diện người – máy.
- Bước 5: Thiết kế hệ thống về mặt kỹ thuật.



3.3.2.3. Mô hình đa cách nhìn - Multiview (4)

- Ưu điểm của mô hình đa cách nhìn:
 - Chỉ rõ các thành phần phải được thực hiện trong quá trình thiết kế giao tiếp người dùng – máy tính.
 - Cung cấp nhiều định hướng cho người thiết kế hệ thống.
 - Nhấn mạnh thứ tự các hoạt động phải tiến hành.

3.3.2.4. Mô hình thiết kế hợp tác (1)

- Thiết kế hợp tác là thiết kế theo ngữ cảnh công việc.
- Người sử dụng không chỉ là người thử nghiệm hệ thống mà còn là thành viên của đội thiết kế.
- Thiết kế hợp tác nhằm mục đích cải tiến các yêu cầu hệ thống thông qua một quy trình thiết kế mang tính lặp, trong đó người sử dụng tham gia một cách tích cực vào quy trình thiết kế.

3.3.2.4. Mô hình thiết kế hợp tác (2)

- 3 đặc điểm quan trọng của thiết kế hợp tác:
 - Cải tiến môi trường làm việc và nhiệm vụ thông qua việc giới thiệu thiết kế. Nhờ đó, việc thiết kế và đánh giá mang tính hướng môi trường hoặc hướng công việc hơn là hướng hệ thống.
 - Thiết kế hợp tác được đặc trưng bởi sự hợp tác: người sử dụng tham gia vào đội thiết kế và có thể đóng góp vào mọi giai đoạn của quá trình thiết kế.
 - Cách tiếp cận của thiết kế hợp tác mang tính lặp: thiết kế có thể được đánh giá và chỉnh sửa lại ở mỗi giai đoạn.

3.3.2.4. Mô hình thiết kế hợp tác (3)

- Một số phương pháp sử dụng trong quy trình thiết kế hợp tác để trao đổi thông tin giữa người sử dụng và người thiết kế:
 - Phương pháp trí tuệ nhóm (Brainstorming): mọi thành viên tham gia vào quy trình thiết kế đều đưa ra ý tưởng. Mọi thông tin được ghi lại, sau đó sử dụng các kỹ thuật khác nhau để chọn lọc ý tưởng.
 - Phương pháp xây dựng bảng tình tiết (Storyboarding): miêu tả các hoạt động thường ngày của người sử dụng cũng như các thiết kế có khả năng và những tác động mà các thiết kế đó sẽ gây ra.

3.3.2.4. Mô hình thiết kế hợp tác (4)

- Một số phương pháp sử dụng trong quy trình thiết kế hợp tác để trao đổi thông tin giữa người sử dụng và người thiết kế (tiếp):
 - Hội thảo (Workshop): dùng để chất vấn lẫn nhau nhằm mục đích cho các bên tham gia có thể hiểu hơn về ngữ cảnh thiết kế từ quan điểm của mỗi thành viên.
 - Xây dựng phác thảo (Pencil and paper exercises): người sử dụng có thể biết được mọi nhiệm vụ bằng cách sử dụng bản phác thảo nháp của thiết kế hệ thống. Điều này chỉ rõ sự khác nhau giữa các yêu cầu của người dùng và thiết kế thực tế.

3.3.3. Mô hình nhận thức (1)

- Mô hình nhận thức nhằm mục đích biểu diễn người dùng khi họ tương tác với giao diện.
- Nhằm mô hình hóa một số sắc thái của người dùng: tri thức, ý định, cách xử lý,...
- Có thể mô tả từ mô hình của nhiệm vụ ở mức độ cao và kết quả của các hoạt động giải quyết vấn đề đến mô tả các hành động ở mức thao tác.

3.3.3. Mô hình nhận thức (2)

- Phân loại theo cách thức mô tả các đặc trưng về khả năng và hiệu năng của người dùng: có 2 loại mô hình.
 - Mô hình khả năng: dự đoán một dãy kết quả các hành vi hợp lệ có thể được thực hiện bởi người dùng.
 - Mô hình hiệu năng: không chỉ miêu tả những kết quả hành vi cần thiết mà còn thực sự miêu tả những gì người sử dụng cần phải biết và cách họ áp dụng kiến thức đó vào việc thực thi các nhiệm vụ thực sự.

3.3.3. Mô hình nhận thức (3)

- Phân loại theo khả năng hiểu biết về người sử dụng, theo ngôn ngữ nhiệm vụ của họ, và theo sự phù hợp giữa ngôn ngữ nhiệm vụ và ngôn ngữ đầu vào: có 3 nhóm mô hình.
 - Mô hình phân cấp mục tiêu và nhiệm vụ GOMS.
 - Mô hình ngôn ngữ.
 - Mô hình mức vật lý Keystroke.

Chương 3 – THIẾT KẾ TƯƠNG TÁC NGƯỜI - MÁY

3.1. Các nguyên tắc khi thiết kế tương tác người - máy

3.2. Quy trình thiết kế

3.3. Mô hình người dùng

3.4. Phân cấp mục tiêu và nhiệm vụ

3.5. Mô hình ngôn ngữ

3.6. Mô hình mức vật lý

Chương 3 – THIẾT KẾ TƯƠNG TÁC NGƯỜI - MÁY

3.4. Phân cấp mục tiêu và nhiệm vụ

3.4.1. *Tổng quan về phân cấp mục tiêu và nhiệm vụ*

3.4.2. *Mô hình GOMS*

3.4.3. *Mô hình CCT*

3.4.1. Tổng quan về phân cấp mục đích và nhiệm vụ (1)

- Người sử dụng đạt được mục đích bằng cách chia mục đích chính thành các mục đích nhỏ hơn và giải quyết từng mục đích nhỏ.
- Kết quả của quá trình là một sơ đồ phân cấp mục đích và các mục đích con của nó.

3.4.1. Tổng quan về phân cấp mục tiêu và nhiệm vụ (2)

- Một số khái niệm:
 - Đích (Goal): là cái mà người dùng mong muốn thực hiện được. Đích gồm nhiều đích con (gọi là phân cấp).
 - Phân cấp: chia thành nhiều mức độ (bắt đầu, giải quyết nhiệm vụ, kết thúc). Có nhiều cách phân cấp, do có nhiều cách để đạt một mục đích.
 - Các kỹ thuật phân cấp:
 - GOMS: Goals – Operator – Methods – Selection
 - CCT : Cognitive Complexity Theory

3.4.1. Tổng quan về phân cấp mục tiêu và nhiệm vụ (3)

- Một số vấn đề khi sử dụng phương pháp phân cấp mục tiêu và nhiệm vụ:
 - Xác định điểm dừng của quá trình phân chia? Nên phân chia nhiệm vụ đến tận mức vận động của người sử dụng hay dừng ở một mức trừu tượng hơn.
 - Xác định điểm bắt đầu của quá trình phân chia? Có thể bắt đầu phân chia từ các điểm khác nhau trên cây phân cấp mục tiêu.
 - Có thể mở rộng phân tích sang các mục đích lớn hơn không?
 - Xử lý lỗi: sơ đồ phân cấp mục tiêu chỉ ra cách đạt mục đích của người dùng hoàn thiện, ít có khả năng dự đoán lỗi.

3.4.2. Mô hình GOMS (1)

- GOMS: **G**oals – **O**perator – **M**ethods and **S**election.

Mục tiêu – Thao tác – Phương thức và Luật lựa chọn.

- GOMS là mô hình để miêu tả phản ứng, tạo ra tính tương thích với chủ thể con người.
- GOMS cho phép mô hình hóa sự phản ứng ở nhiều cấp độ trừu tượng, từ nhiệm vụ đến các hành động mức vật lý.
- Phân cấp mục tiêu và nhiệm vụ GOMS xây dựng nên cấu trúc hình thức, cho phép tổ chức quá trình quan niệm.

3.4.2. Mô hình GOMS (2)

- Phương thức quan niệm được đánh giá theo hai hướng: phân tích nhiệm vụ và ước đoán phản ứng của người dùng khi hoàn thành nhiệm vụ.
- GOMS mô tả một tập các phương thức cần thiết để thực hiện mục đích, bao gồm một dãy các thao tác mà người dùng có thể lựa chọn để thực hiện một mục đích con.
- GOMS cũng cung cấp luật lựa chọn nếu có nhiều hơn một phương thức để chọn.

3.4.2. Mô hình GOMS (3) – Bốn thành phần cơ bản của GOMS

- Goals – Mục tiêu:
 - Là mục đích mà người dùng muốn thực hiện.
 - Là một cấu trúc ký pháp định nghĩa một trạng thái mong muốn.
 - Các mục tiêu thường được chia nhỏ thành các đích con (mục tiêu cơ sở).
 - Để đạt được mục tiêu chính, người dùng phải thực hiện được các đích con.
 - Một mục tiêu cơ sở đạt được khi thực hiện một dãy các thao tác.
 - Các mục tiêu thường được sắp xếp theo thứ tự phân cấp (không bắt buộc); và ở dạng *hành động* – *đối tượng*.

3.4.2. Mô hình GOMS (4) – Bốn thành phần cơ bản của GOMS

- Operator – Thao tác:

- Là các thao tác cơ bản của người dùng như: nhấn phím, di chuyển chuột, suy nghĩ,... nhằm thay đổi trạng thái (trạng thái tâm lý của người dùng hay trạng thái môi trường).

- Là mức thấp nhất của quá trình phân tích.

- Là hành động mà người dùng dùng để thực hiện nhiệm vụ.

- Cũng ở dạng *hành động – đối tượng*

- Thông số quan trọng của hành động: thời gian thực hiện.

- Thời gian thực hiện có thể là một hằng số hoặc không phải hằng số.

3.4.2. Mô hình GOMS (5) – Bốn thành phần cơ bản của GOMS

- **Operator – Thao tác:** Hai dạng của thao tác:
 - **Hành động ngoài:** Là các hành động có thể quan sát được. Người dùng sử dụng để trao đổi thông tin với hệ thống hay với các đối tượng khác của môi trường. Hành động ngoài thường bao gồm các vận động cảm giác, vận động hay tương tác.

Ví dụ: hành động ấn phím.

- **Hành động trong:** thuộc các mức suy nghĩ của người dùng, không quan sát được. Chỉ tác động đến trạng thái tinh thần của người sử dụng.

Ví dụ: hành động đọc hộp hội thoại.

3.4.2. Mô hình GOMS (6) – Bốn thành phần cơ bản của GOMS

- **METHOD** – Phương thức:
 - Phân rã mục tiêu thành các mục đích con/ thao tác con.
 - Phương thức là một chuỗi các bước và các đích con cần thực hiện để đạt mục tiêu.
 - Phương thức mô tả cách thức để đạt mục tiêu.
 - Trong GOMS, mục đích phân chia được xem là phương thức.
 - Một bước trong phương thức thường bao gồm các thao tác (các hành động ngoài và hành động trong).
 - Có nhiều phương thức để thực hiện một nhiệm vụ.

3.4.2. Mô hình GOMS (7) – Bốn thành phần cơ bản của GOMS

- **SELECTION RULE** – Luật lựa chọn:
 - Là cách thức chọn lấy một phương thức phù hợp để hoàn thành mục đích.
 - GOMS không cho phép chọn các phương thức một cách ngẫu nhiên, mà cố gắng dự đoán các phương thức sẽ được sử dụng.
 - Luật lựa chọn phụ thuộc vào người sử dụng, vào trạng thái của hệ thống và vào các chi tiết của mục đích.
 - Luật lựa chọn thường được mô tả dưới dạng IF ... THEN ...

3.4.2. Mô hình GOMS (8) – Ưu nhược điểm của GOMS

- GOMS là mô hình hiệu năng.
- GOMS cung cấp cho nhà thiết kế những hỗ trợ hình thức để đánh giá trước hiệu năng.
- GOMS không hỗ trợ lý thuyết để cấu trúc hóa nhiệm vụ.
- GOMS quá giản lược.
- GOMS chỉ ra quá trình thực hiện nhiệm vụ với giả thiết là không có lỗi. Tuy nhiên, lỗi là điều khó tránh.

3.4.3. Mô hình CCT (1)

- CCT: **Cognitive Complexity Theory**

Lý thuyết độ phức tạp nhận thức.

- Phát triển dựa trên quá trình phân chia mục đích trong mô tả GOMS, có khả năng dự đoán tốt hơn GOMS.
- CCT là hai mô tả song song: về mục đích của người sử dụng và về hệ thống máy tính.
- Mô tả mục đích người sử dụng: giống mô hình phân cấp GOMS. Sử dụng luật sản xuất để mô tả.
- Mô tả hệ thống: sử dụng mạng chuyển hóa trạng thái suy rộng, một dạng của mạng chuyển hóa trạng thái.

3.4.3. Mô hình CCT (2)

- Luật sản xuất: là một chuỗi luật dạng IF *điều kiện* THEN *hành động*.
- Điều kiện: là một phát biểu về các nội dung của bộ nhớ làm việc.
- Nếu điều kiện đúng: luật sản xuất được cháy.
- Hành động: có thể gồm một hoặc nhiều hành động cơ bản.
- Hành động có thể làm thay đổi bộ nhớ làm việc hoặc chỉ là các hành động ngoài (như bấm phím).

Chương 3 – THIẾT KẾ TƯƠNG TÁC NGƯỜI - MÁY

3.1. Các nguyên tắc khi thiết kế tương tác người - máy

3.2. Quy trình thiết kế

3.3. Mô hình người dùng

3.4. Phân cấp mục tiêu và nhiệm vụ

3.5. Mô hình ngôn ngữ

3.6. Mô hình mức vật lý

Chương 3 – THIẾT KẾ TƯƠNG TÁC NGƯỜI - MÁY

3.5. Mô hình ngôn ngữ

3.5.1. Ký pháp BNF

3.5.2. Văn phạm nhiệm vụ – hành động TAG

3.5. Mô hình ngôn ngữ

- Tương tác người dùng – máy tính thường được xem xét dưới góc độ ngôn ngữ.
- Nhiều mô hình hình thức được phát triển xung quanh khái niệm ngôn ngữ.
- Hai mô hình ngôn ngữ tiêu biểu:
 - Ký pháp văn phạm BNF (Backus Naur Form)
 - Văn phạm nhiệm vụ - hành động TAG (Task – Action – Grammar)

3.5.1. Ký pháp BNF (1)

- BNF: Backus Naur Form: Các luật để mô tả văn phạm đối thoại.
- BNF dùng để đặc tả đối thoại, giúp hiểu được hành vi của người dùng và phân tích các khó khăn về nhận thức của tương tác.
- Bao gồm các luật để mô tả đối thoại, chỉ liên quan đến cú pháp, bỏ qua ngữ nghĩa của ngôn ngữ.
- Thường được sử dụng để đặc tả cú pháp của các ngôn ngữ lập trình máy tính.

3.5.1. Ký pháp BNF (2) – Một số quy ước

- BNF dùng hai loại ký hiệu:
 - Ký hiệu kết thúc: viết bằng chữ in hoa.
 - Ký hiệu không kết thúc: viết bằng chữ in thường.
- Ký hiệu “ $::=$ ” hiểu là “*được định nghĩa*”.
- BNF chỉ biểu diễn hành động của người dùng mà không đề cập đến cảm nhận của người dùng về sự đáp ứng của hệ thống.

3.5.2. Văn phạm nhiệm vụ – hành động TAG (1)

- TAG: Task – Action – Grammar.
- TAG đưa vào một số phần tử như luật văn phạm có tham số.
- Nhằm nhấn mạnh tính nhất quán và mã hóa tri thức người dùng.
- Ví dụ: một số lệnh trong UNIX (đôi số là tệp hoặc thư mục)
 - Sao tệp:
 $cp ::= 'cp' + \text{filename} + \text{filename} / 'cp' + \text{filename} + \text{directory}$
 - Chuyển tệp:
 $mv ::= 'mv' + \text{filename} + \text{filename} / 'mv' + \text{filename} + \text{directory}$
 - Liên kết tệp:
 $lk ::= 'lk' + \text{filename} + \text{filename} / 'lk' + \text{filename} + \text{directory}$

3.5.2. Văn phạm nhiệm vụ – hành động TAG (2)

- Mô tả lệnh trong văn phạm TAG: (với các lệnh cp, mv, lk)

File(Op) := command(Op) + filename + filename
/command(Op) + filename + directory

command(op = cp) := 'cp'

command(op = mv) := 'mv'

command(op = lk) := 'lk'

Chương 3 – THIẾT KẾ TƯƠNG TÁC NGƯỜI - MÁY

3.1. Các nguyên tắc khi thiết kế tương tác người - máy

3.2. Quy trình thiết kế

3.3. Mô hình người dùng

3.4. Phân cấp mục tiêu và nhiệm vụ

3.5. Mô hình ngôn ngữ

3.6. Mô hình mức vật lý

Chương 3 – THIẾT KẾ TƯƠNG TÁC NGƯỜI - MÁY

3.6. Mô hình mức vật lý

3.6.1. *Tổng quan về mô hình KLM*

3.6.2. *Các bước trong mô hình KLM*

3.6.3. *Các thao tác và nguyên tắc tính thời gian thực hiện*

3.6.4. *Một số luật Heuristic*

3.6.5. *Một số ưu nhược điểm của mô hình KLM*

3.6.1. Tổng quan về mô hình KLM (1)

- Mô hình KLM (Keystroke – Level Model): do Card, Moral và Newell đề xuất năm 1983.
- KLM là một mô hình con của GOMS.
- Nhằm đánh giá thời gian thực hiện một nhiệm vụ từ một thiết kế xác định với kịch bản nhiệm vụ cụ thể.
- Liệt kê tuần tự các hành động ở mức nhấn phím mà người dùng phải thực hiện để hoàn thành một công việc, đối chiếu thời gian cần thiết để thực hiện công việc.
- Không cần phải có thiết kế hay chế thử.
- Chỉ yêu cầu mô tả giao diện người dùng chi tiết để có thể thực hiện tuần tự các hành động được yêu cầu.

3.6.1. Tổng quan về mô hình KLM (2)

- Các hành động ở mức nhấn phím: nếu có cùng mức với các hành động như nhấn phím, di chuột, nhấn phím chuột.
- Yêu cầu giao diện người dùng phải chỉ rõ người dùng sẽ thực hiện nhiệm vụ theo từng hành động ở mức nhấn phím như thế nào.
- Các hành động cơ bản được gọi là các **operator**.
- Trong KLM, có một tập các operator chuẩn, với thời gian thực hiện được lượng giá theo dữ liệu thực nghiệm.

3.6.2. Các bước trong mô hình KLM (1)

- Bước 1: Chọn một hay nhiều kịch bản nhiệm vụ tiêu biểu.
- Bước 2: Xác định thiết kế chi tiết tới mức có thể liệt kê các hành động mức nhấn phím cho các kịch bản nhiệm vụ cụ thể.
- Bước 3: Với mỗi kịch bản nhiệm vụ, tìm cách tốt nhất để hoàn thành hoặc chọn cách được cho là người dùng sẽ sử dụng.
- Bước 4: Liệt kê các hành động mức phím nhấn hay mức vật lý tương ứng với việc thực hiện nhiệm vụ.

3.6.2. Các bước trong mô hình KLM (2)

- Bước 5: Thêm các hành động suy nghĩ khi người dùng dừng lại hay suy nghĩ.
- Bước 6: Xác định thời gian thực hiện chuẩn cho mỗi hành động.
- Bước 7: Nếu cần, có thể thêm các hành động mà người dùng phải chờ sự đáp ứng của hệ thống.
- Bước 8: Thời gian thực hiện nhiệm vụ được lượng giá bằng tổng thời gian thực hiện của tất cả các hành động.

3.6.2. Các bước trong mô hình KLM (3) – Các giả thiết

- Một nhiệm vụ có thể phân tích thành các nhiệm vụ con.
- Có ngôn ngữ để điều khiển hành động.
- Có các thông số về khả năng vận động của người dùng.
- Có các thông số về khả năng đáp ứng của hệ thống.
- Có các phương thức để thực hiện nhiệm vụ.
- Người dùng có kinh nghiệm.

3.6.3. Các thao tác và nguyên tắc tính thời gian thực hiện (1)

- 6 thao tác mô tả việc thực hiện một nhiệm vụ cơ bản trong KLM:
 - K – Keystroking: Gõ phím hay nhấn chuột.
 - P – Pointing: Di chuyển con trỏ chuột tới một điểm trên màn hình.
 - H – Homing: Thay đổi công cụ.
 - D – Drawing: Hành động vẽ.
 - M – Mental activity: Hành động suy nghĩ.
 - R – Respond Time: Thời gian đáp ứng của hệ thống.
- Hai hành động phụ:
 - B: hành động nhấn đơn trên phím chuột (khoảng 0.1s).
 - BB: hành động nhấn kép phím chuột (khoảng 0.2s).

3.6.3. Các thao tác và nguyên tắc tính thời gian thực hiện (2)

- Tổng thời gian thực hiện một thao tác:

$$T_{\text{exe}} = T_K + T_P + T_H + T_D + T_M + T_R$$

- K: hành động nhấn phím trên bàn phím hay phím chuột.

T_K : từ 0.28s – 1.2s, trung bình khoảng 0.75s.

Người đánh máy chuyên nghiệp: 0.12s

Người không quen bàn phím: 1.2s

Thời gian gõ lần lượt n phím:

$$T(n) = n \times T_K$$

3.6.3. Các thao tác và nguyên tắc tính thời gian thực hiện (3)

$$T_{\text{exe}} = T_K + T_P + T_H + T_D + T_M + T_R$$

- P: hành động di chuyển con trỏ tới đích.

T_P : được tính theo luật Fitts.

T_P : 0.8s – 1.5s. Trung bình giá trị này là 1.1s

- H: hành động thay đổi công cụ (từ bàn phím sang chuột và ngược lại).

T_H : 0.4s

3.6.3. Các thao tác và nguyên tắc tính thời gian thực hiện (4)

$$T_{\text{exe}} = T_K + T_P + T_H + T_D + T_M + T_R$$

- D: hành động vẽ dùng chuột.

T_D : phụ thuộc vào số lượng nét vẽ.

$T_D = 0.9n + 0.161$ với n là số nét vẽ.

- M: hoạt động tâm lý người dùng cần có khi thực hiện các thao tác vật lý K, P, H hay D.

T_M : 0.6s – 1.35s. Trung bình là khoảng 1.2s

- R: Thời gian chờ đợi sự phản hồi của hệ thống.

3.6.4. Một số luật Heuristic (1)

- Được áp dụng để thêm vào hay loại bỏ đi một số M , do M là tham số với người dùng, phụ thuộc vào các hành động vật lý và khả năng của người dùng.
- *Luật 0*: Đặt M trước tất cả các K không phải là vị trí bắt đầu của dãy. Đặt M trước P nếu tương đương với việc chỉ ra tên một điều khiển.
- *Luật 1*: Lược bỏ M nếu thao tác tiếp theo có thể dự đoán khi đang thực hiện các thao tác trước M .

3.6.4. Một số luật Heuristic (2)

- *Luật 2:* Nếu dãy có dạng MKMK...MK thì loại bỏ tất cả các M, trừ M đầu tiên.
- *Luật 3:* Nếu K là một ký tự kết thúc rườm rà (là ký tự theo sau một ký hiệu báo sự kết thúc khác) thì loại bỏ M trước K.
- *Luật 4:* Nếu K xác định một hằng số thì bỏ M, ngược lại khi K xác định một biến số thì cần giữ lại M.

3.6.5. Một số ưu nhược điểm của mô hình KLM

- KLM là một công cụ phân tích định lượng, cho phép so sánh và tiên đoán hiệu năng với người dùng trong một cú pháp cho trước.
- KLM khá đơn giản, hệ thống thuật ngữ và kỹ thuật mô hình khá dễ hiểu.
- Việc thay các M trong KLM không đơn giản.
- Mô hình KLM không tính đến thời gian đáp ứng của hệ thống, nên không cụ thể được việc lượng giá thời gian.
- Không có sự phân biệt cụ thể giữa các phím với nhau (phím thông dụng và phím đặc biệt).
- Chỉ quan tâm đến thời gian thực hiện, không quan tâm đến tổng thời gian hoàn thành nhiệm vụ.

Chương 4 – MÔ HÌNH HỆ THỐNG

4.1. Quy trình thiết kế hệ tương tác

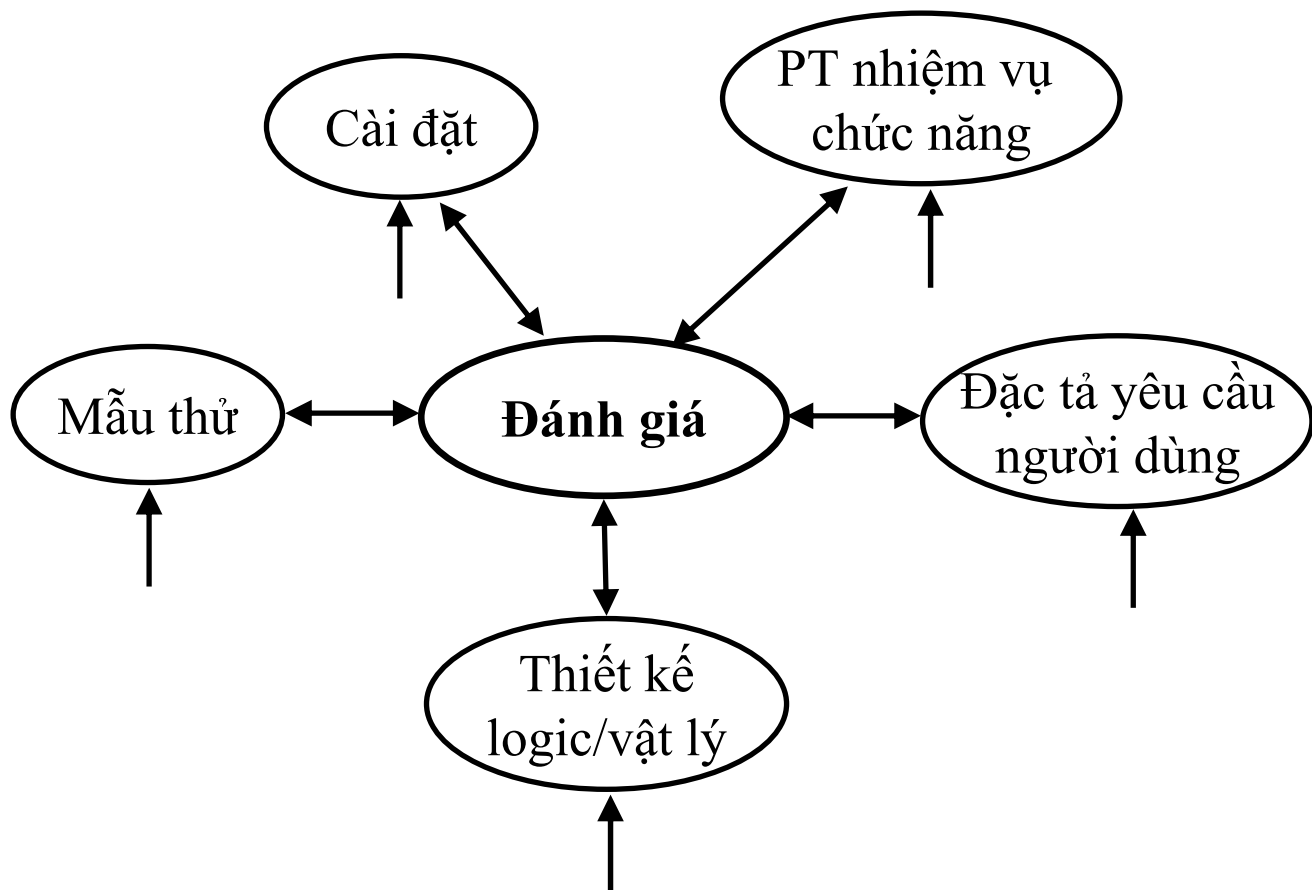
4.2. Đặc tả yêu cầu người dùng

4.3. Phân tích nhiệm vụ

4.4. Mô hình thoại

4.1. Quy trình thiết kế tương tác (1)

- Thiết kế tương tác nhằm hỗ trợ con người trong công việc cũng như trong cuộc sống.
- Vòng đời hình sao:



Chương 4 – MÔ HÌNH HỆ THỐNG

4.1. Quy trình thiết kế hệ tương tác

4.2. Đặc tả yêu cầu người dùng

4.3. Phân tích nhiệm vụ

4.4. Mô hình thoại

Chương 4 – MÔ HÌNH HỆ THỐNG

4.2. Đặc tả yêu cầu người dùng

4.2.1. *Khái niệm*

4.2.2. *Đặc tả chức năng*

4.2.3. *Đặc tả dữ liệu*

4.2.4. *Đặc tả tính tiện dụng*

4.2.1. *Khái niệm (1)*

- Đặc tả yêu cầu người dùng là quá trình thu thập các yêu cầu người dùng đối với hệ thống cần phát triển.
- Đây là bước đầu tiên và là một trong hai giai đoạn quan trọng nhất trong phát triển HCI.
- Nhiệm vụ của khách hàng: cung cấp bản mô tả về những yêu cầu của mình, điều mà họ mong muốn hệ thống mới phải cung cấp được.
- Cung cấp của khách hàng là một bản mô tả còn chung chung.

4.2.1. *Khái niệm (2)*

- Nhiệm vụ của người thiết kế: cần làm rõ bản mô tả của khách hàng và xác định:
 - Những điểm không cần thiết
 - Những điểm không thể thực hiện được hoặc không chắc chắn thực hiện được
 - Những điểm còn thiếu
 - Những điểm nhập nhằng hay mâu thuẫn

4.2.1. *Khái niệm (3)*

- Các kỹ thuật sử dụng:
 - Quan sát
 - Phỏng vấn
 - Phân tích tài liệu
- Kết quả của bước đặc tả yêu cầu người dùng:
 - Biểu diễn bài toán với hệ thống hiện tại
 - Biểu diễn các yêu cầu của hệ thống mới

4.2.1. *Khái niệm (4)*

- Các kiểu đặc tả yêu cầu người dùng:
 - Đặc tả chức năng
 - Đặc tả dữ liệu
 - Đặc tả tính tiện dụng

4.2.2. Đặc tả chức năng (1)

- Đặc tả chức năng: là đặc tả nhiệm vụ mà hệ thống (con người, máy tính, ...) phải làm.
- Đặc tả chức năng không quyết định hoạt động nào được thực hiện bởi máy tính, hoạt động nào được thực hiện bởi con người. Việc này được quyết định sau bước phân tích nhiệm vụ.
- Đặc tả chức năng còn bao gồm đặc tả các ràng buộc phải xét đến khi thực hiện các chức năng.

4.2.2. *Đặc tả chức năng (2)*

- Đặc tả chức năng thường được chia thành nhiều môđun, được tổ chức theo kiểu phân cấp (từ mức trừu tượng đến mức cụ thể).
- Mục đích phân cấp: việc mô tả hệ thống được chia thành nhiều phần, do đó dễ điều khiển và xử lý riêng biệt từng phần.
- Không thể thu thập và đặc tả các yêu cầu chức năng chỉ trong một lần.

4.2.2. Đặc tả chức năng (3)

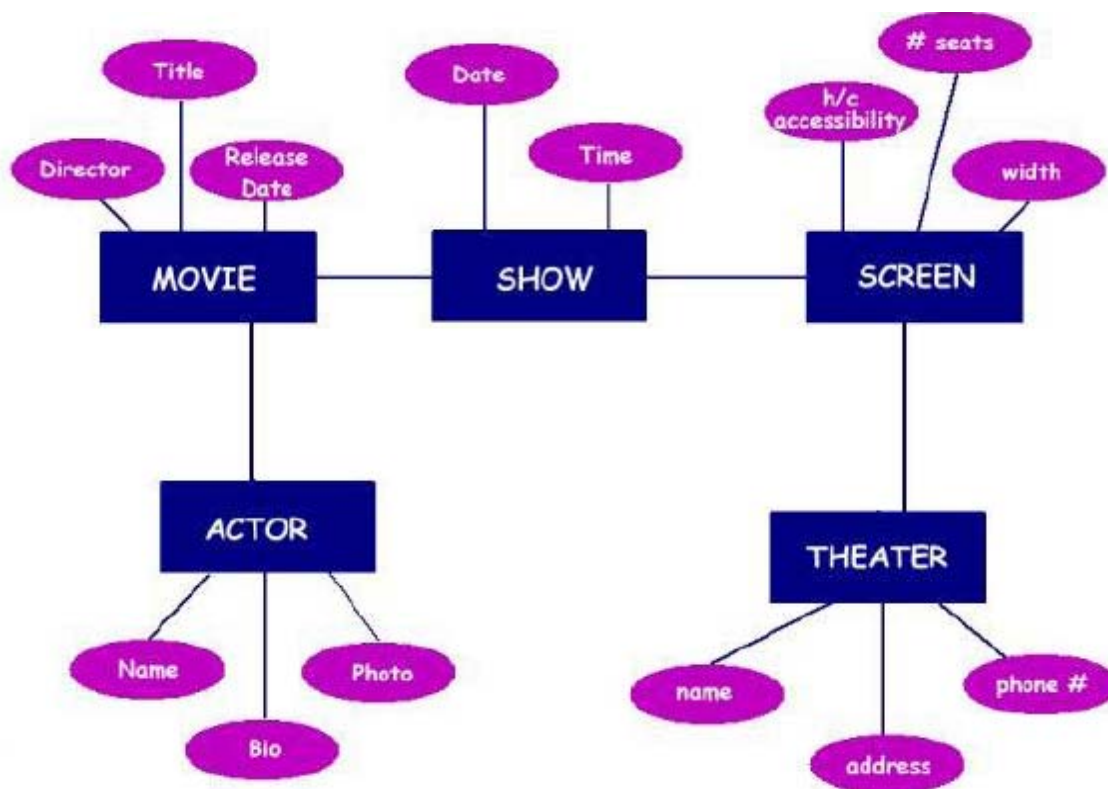
- Một số công cụ đặc tả chức năng:
 - Sơ đồ luồng dữ liệu (Data Flow Diagram – DFD)
 - Máy trạng thái hữu hạn
 - Ngôn ngữ tự nhiên có cấu trúc

4.2.3. Đặc tả dữ liệu (1)

- Đặc tả dữ liệu: là biểu diễn luồng dữ liệu, ngữ nghĩa và cấu trúc của dữ liệu đáp ứng yêu cầu của người dùng.
- Khác với đặc tả chức năng: đặc tả dữ liệu chỉ tập trung vào cấu trúc dữ liệu.
- Đặc tả dữ liệu được liệt kê nhờ các kỹ thuật:
 - Quan sát
 - Phỏng vấn
 - Phân tích tài liệu
- Điều quan trọng là phải hiểu và định nghĩa một cách chính xác các phần tử dữ liệu.

4.2.3. Đặc tả dữ liệu (2)

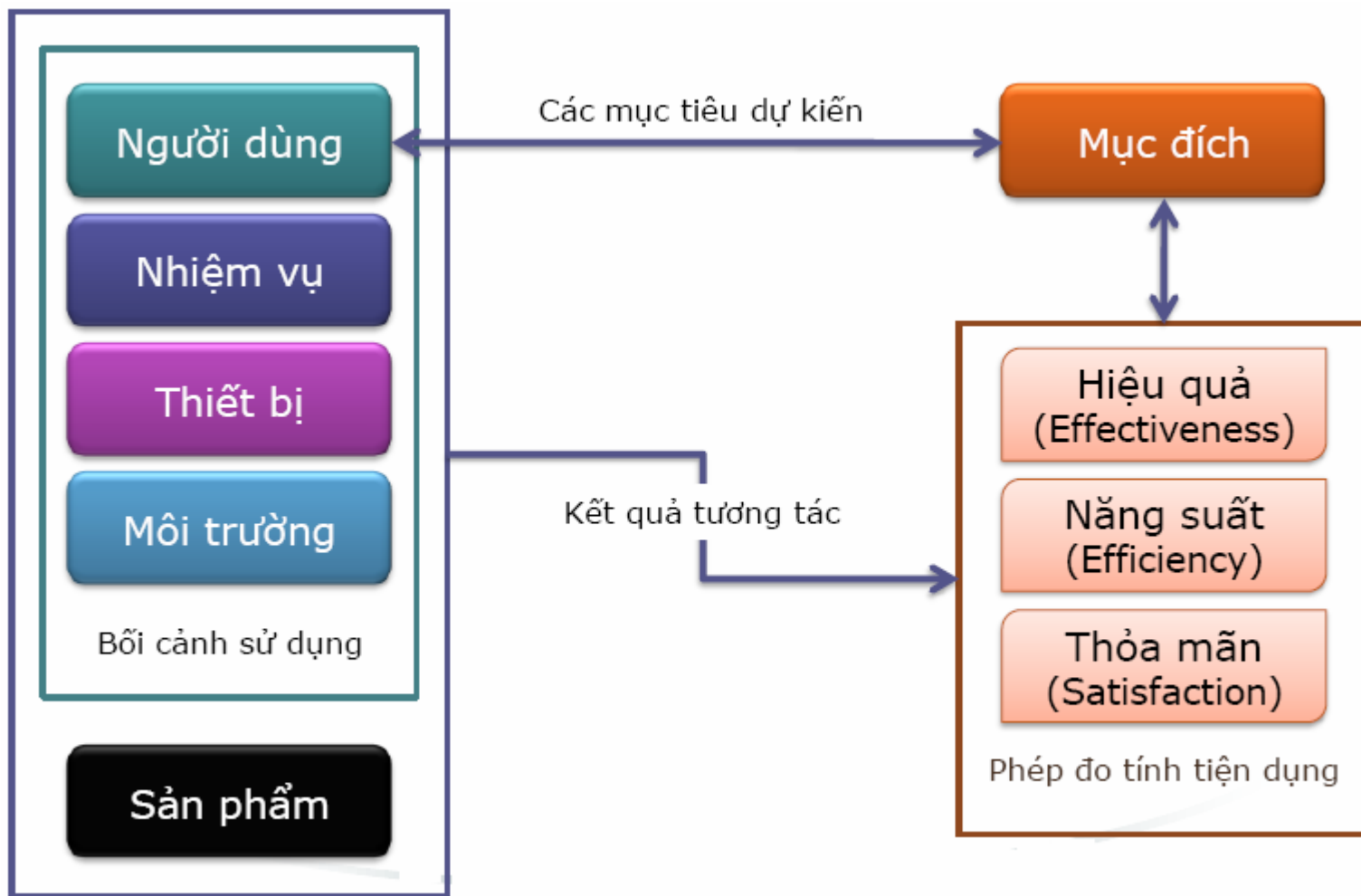
- Công cụ đặc tả dữ liệu: Lưu đồ thực thể liên kết (Entity Relationship Diagram – ERD).



4.2.4. Đặc tả tính tiện dụng (1)

- Trong công nghệ phần mềm và hệ thống thông tin: chỉ quan tâm tới đặc tả chức năng và đặc tả dữ liệu.
- Trong HCI, cần quan tâm thêm đặc tả tính tiện dụng.
- Đặc tả tính tiện dụng được tiến hành đồng thời với đặc tả chức năng và đặc tả dữ liệu.
- Các kỹ thuật sử dụng:
 - Quan sát
 - Phỏng vấn
 - Phân tích tài liệu
- Đặc tả tính tiện dụng liên quan đến đánh giá quy trình thiết kế.

4.2.4. Đặc tả tính tiện dụng (2) – Khung tính tiện dụng (chuẩn ISO 9241 -11)



Chương 4 – MÔ HÌNH HỆ THỐNG

4.1. Quy trình thiết kế hệ tương tác

4.2. Đặc tả yêu cầu người dùng

4.3. Phân tích nhiệm vụ

4.4. Mô hình thoại

Chương 4 – MÔ HÌNH HỆ THỐNG

4.3. Phân tích nhiệm vụ

4.3.1. Vai trò của phân tích nhiệm vụ

4.3.2. Một số thuật ngữ

4.3.3. Phân tích nhiệm vụ phân cấp

4.3.4. Phân tích nhiệm vụ theo nhận thức

4.3.5. Phân tích nhiệm vụ theo mô hình tri thức

4.3.6. Sử dụng kỹ thuật phân tích nhiệm vụ

4.3.1. Vai trò của phân tích nhiệm vụ (1)

- Phân tích nhiệm vụ: là quá trình thu thập dữ liệu về những nhiệm vụ mà người sử dụng thực hiện và phân tích chúng để có được sự hiểu biết sâu sắc về các nhiệm vụ đó.
- Phân tích nhiệm vụ tập trung tìm hiểu:
 - Người dùng.
 - Đối tượng mà người dùng tác động vào.
 - Hiểu biết mà người dùng cần có.

4.3.1. Vai trò của phân tích nhiệm vụ (2)

- Quá trình phân tích nhiệm vụ phải nắm được:
 - Cách thức con người hiểu công việc (hay đích phải thực hiện).
 - Các đối tượng mà người dùng thao tác trên đó.
 - Những tri thức mà người dùng cần có để thực thi nhiệm vụ, nhằm đạt được đích mong muốn.

4.3.1. Vai trò của phân tích nhiệm vụ (3) – Ví dụ

Phân tích nhiệm vụ “Hút bụi”

- **Mục đích:** Hút bụi trong nhà.
- **Các công việc cần làm:**
 - Lấy máy hút bụi.
 - Lắp các phụ tùng cần thiết.
 - Thực hiện hút bụi.
- **Một số điều kiện:**
 - Khi hộp rác đầy: tháo bỏ hộp rác, đổ rác và lắp lại.
 - Khi hút xong: tháo các phụ tùng và cất máy hút bụi.

4.3.1. Vai trò của phân tích nhiệm vụ (4) – Ví dụ

- **Tri thức cần có:**
 - Cách sử dụng máy hút bụi.
 - Cách tháo lắp các chi tiết.
 - Trình tự hút bụi ở các phòng.

4.3.2. Một số thuật ngữ (1)

- Mục đích (Goal):
 - Còn được gọi là nhiệm vụ ngoài.
 - Là một trạng thái của hệ thống mà người dùng muốn hoàn thành.
 - Mục đích có thể được mô tả ở mức độ trừu tượng.
 - Các thiết bị thực hiện mục đích là các công cụ, phương pháp, tác nhân kỹ thuật có thể làm thay đổi trạng thái hệ thống.

Ví dụ: Mục đích: viết thư

Các phương tiện: giấy, bút; máy tính; máy soạn thảo văn bản,...

- Mục đích có thể gồm nhiều đích con hay nhiệm vụ (nhiệm vụ trong).

4.3.2. Một số thuật ngữ (2)

- **Nhiệm vụ (Task):**
 - Là cái người dùng cần làm để thực hiện mục đích đặt ra.
 - Là một tập có cấu trúc các hoạt động, trong đó các hành động được làm theo một trình tự nào đó.
- **Hành động (Action):**
 - Là một nhiệm vụ:
 - + Không bao hàm việc giải quyết vấn đề
 - + Không có cấu trúc điều khiển

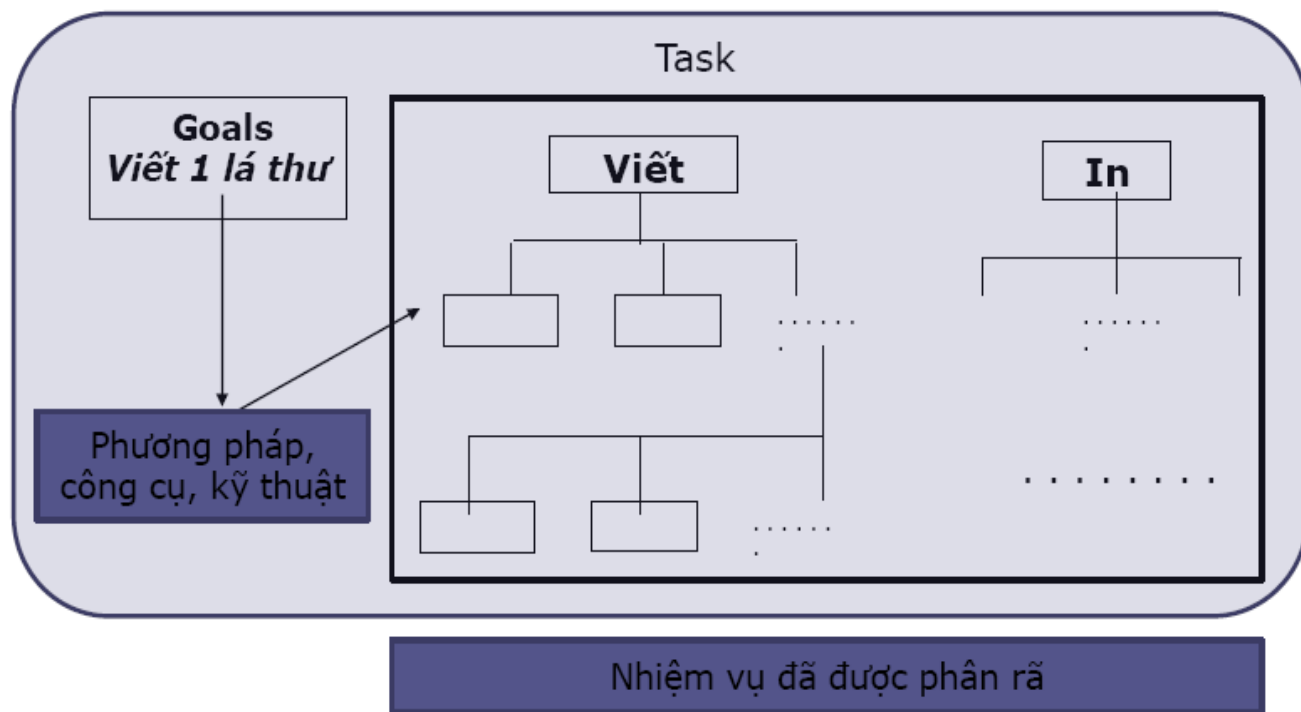
4.3.2. Một số thuật ngữ (3) – Ví dụ: Viết thư với MS Word

- Mục đích:

Viết một lá thư

- Thiết bị:

Các phương pháp, công cụ, kỹ thuật sử dụng để viết thư (máy tính, MS Word)



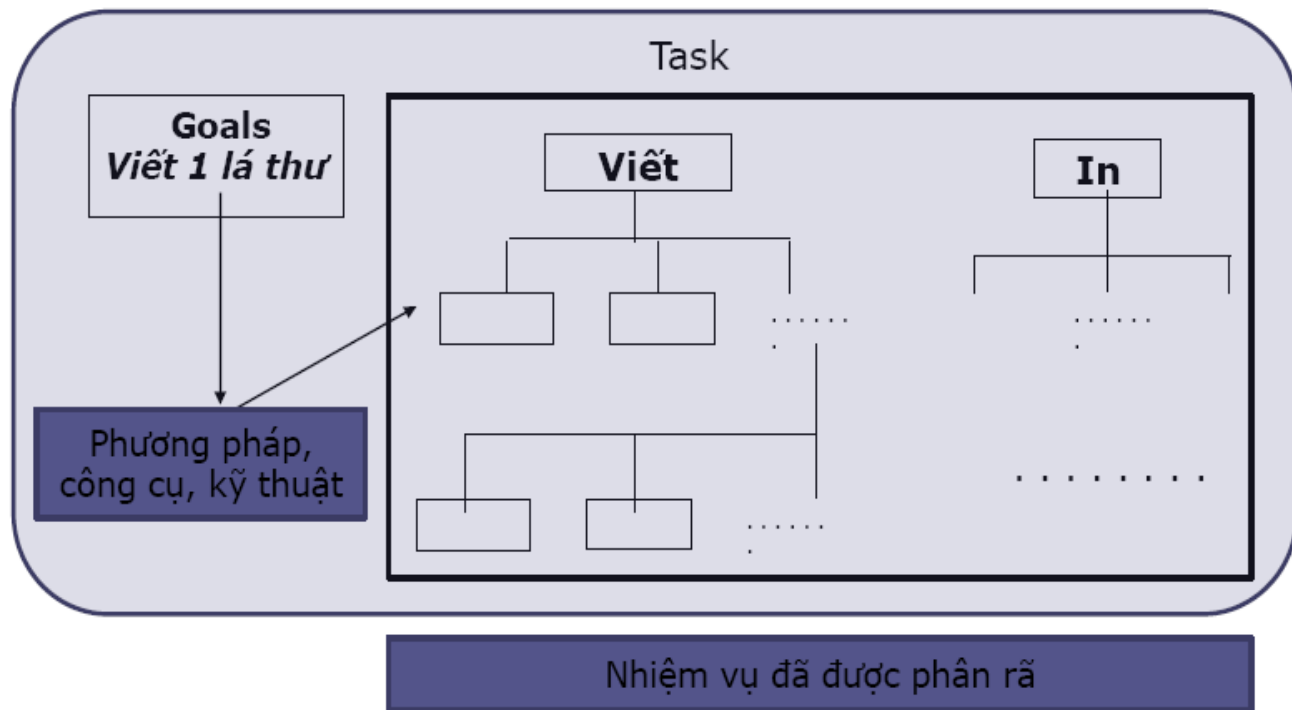
4.3.2. Một số thuật ngữ (4) – Ví dụ (tiếp)

- Nhiệm vụ:

Các hoạt động cần thiết để hoàn thành mục đích bằng các thiết bị.

- Viết thư

- In thư



4.3.2. Một số thuật ngữ (5) – Ví dụ (tiếp)

- Nhiệm vụ con:

Các thành phần của nhiệm vụ

- Viết thư:

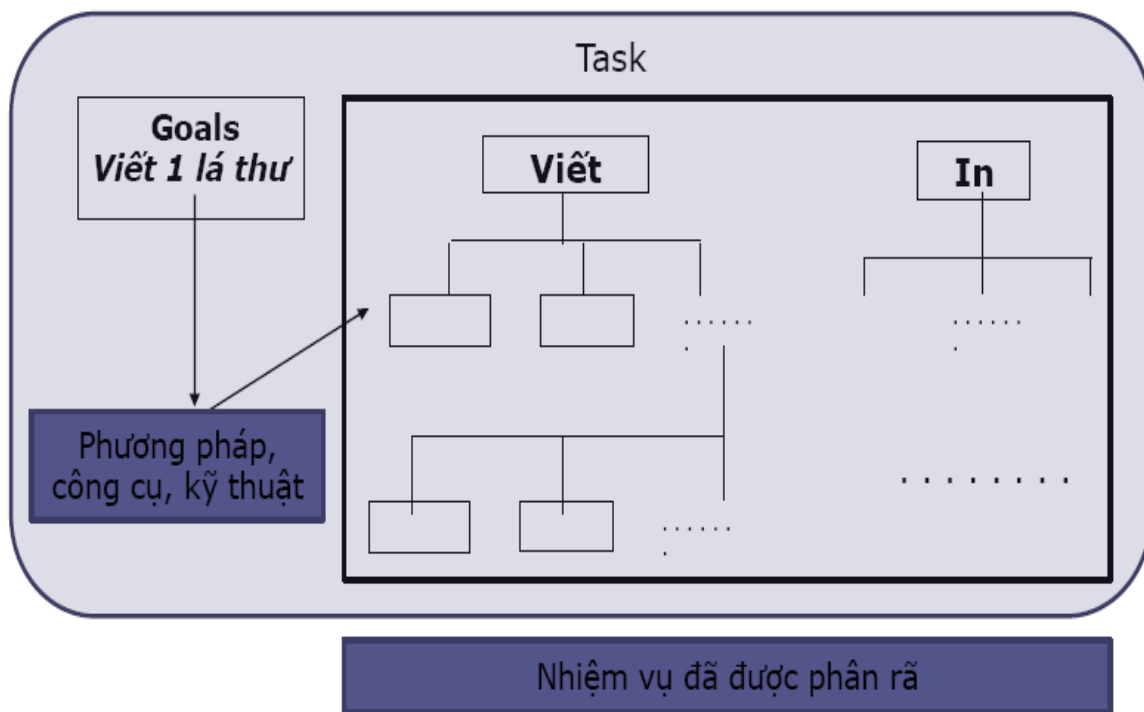
+ Khởi tạo MS Word

+ Nhập dữ liệu

+ Định dạng văn bản

+ Hiệu chỉnh

+ Lưu văn bản



4.3.2. Một số thuật ngữ (6) – Ví dụ (tiếp)

- Hành động:

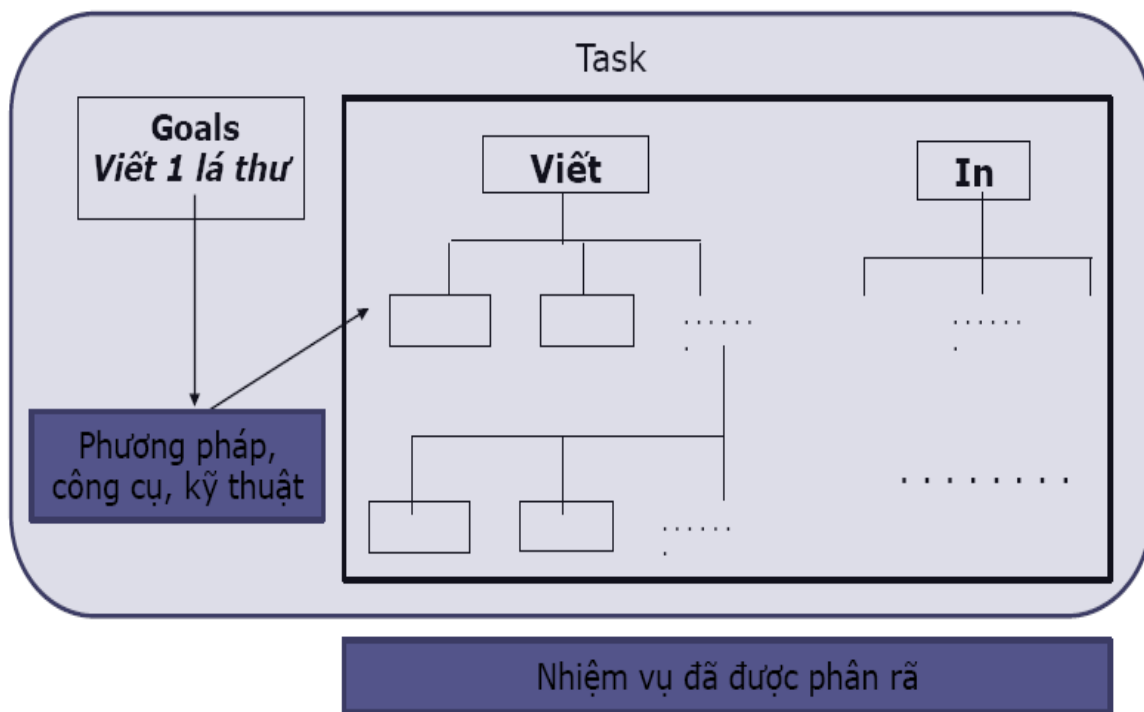
Các nhiệm vụ đơn giản không có cấu trúc điều khiển

- Nhập dữ liệu:

+ Nhập từ bàn phím

+ Nạp từ tệp

+ Phương pháp khác



4.3.3. Phân tích nhiệm vụ phân cấp (1)

- Viết tắt là HTA (Hierarchical Task Analysis).
- Mô tả các nhiệm vụ của hệ thống dưới dạng các thao tác và các kế hoạch.
- Đầu ra là một cây phân cấp các nhiệm vụ hay các nhiệm vụ con và kế hoạch miêu tả trình tự và điều kiện thực hiện các nhiệm vụ.
- Thao tác là những hoạt động mà người dùng thực hiện để đạt được một mục đích nào đó.
- Kế hoạch là các ràng buộc điều kiện về thời điểm thực hiện thao tác.
- Các mức thao tác không theo thứ tự, kế hoạch chỉ ra thứ tự thực hiện các thao tác.

4.3.3. Phân tích nhiệm vụ phân cấp (2)

- Một số loại kế hoạch thường gặp:
 - Thứ tự cố định: các nhiệm vụ con được thực hiện theo đúng thứ tự mà chúng được phân chia.
 - Các nhiệm vụ lựa chọn: việc thực hiện hay không thực hiện nhiệm vụ tùy thuộc vào từng tình huống cụ thể.
 - Chờ đợi các sự kiện: nhiệm vụ chỉ được thực hiện khi một nhiệm vụ liên quan đến phải được hoàn thành trước.
 - Chia sẻ thời gian: các nhiệm vụ được thực hiện đồng thời.
 - Nhiệm vụ không bắt buộc: có thể lựa chọn thực hiện hoặc không thực hiện nhiệm vụ mà vẫn đảm bảo đạt được mục đích.

4.3.3. Phân tích nhiệm vụ phân cấp (3)

- Các cách biểu diễn:
 - Sử dụng ký pháp viết lùi đầu dòng để biểu diễn các mức và đánh số các nhiệm vụ để nhấn mạnh tính phân cấp.
 - Có thể sử dụng ký pháp đồ họa để biểu diễn các mức phân cấp.
 - Nếu quá trình thực hiện nhiệm vụ có phụ thuộc điều kiện thì cần biểu diễn dưới dạng rẽ nhánh, lặp và chờ đợi.
- Cách đánh số:
 - Mục đích thường được đánh số 0.
 - Các nhiệm vụ được đánh số bắt đầu từ 1.
 - Các kế hoạch cũng được đánh số bắt đầu từ 0.

4.3.3. Phân tích nhiệm vụ phân cấp (4) – Ví dụ 1:

Biểu diễn phân tích công việc “Hút bụi” bằng ký pháp viết lùi:

0. Mục đích: Hút bụi trong nhà

1. Lấy máy hút bụi

2. Lắp các phụ tùng

3. Thực hiện hút bụi

3.1. Hút bụi ngoài hành lang

3.2. Hút bụi trong phòng khách

3.3. Hút bụi trong phòng ngủ

4. Đổ túi chứa bụi

5. Tháo các phụ tùng và cất máy

4.3.3. Phân tích nhiệm vụ phân cấp (5) – Ví dụ 1 (tiếp):

Plan 0 (Kế hoạch 0):

- Thể hiện cách thực hiện các nhiệm vụ từ 1 đến 5 của đích 0.
- Thực hiện các công việc 1 – 2 – 3 – 5 theo trình tự.
- Khi túi chứa bụi đầy thì thực hiện công việc 4.

Plan 3 (Kế hoạch 3):

- Thể hiện cách thực hiện các nhiệm vụ từ 3.1 đến 3.3.
- Thực hiện các nhiệm vụ 3.1, 3.2, 3.3 theo thứ tự bất kỳ, phụ thuộc vào nhu cầu.

Các nhiệm vụ 1, 2 và 4: không có kế hoạch thực hiện vì các nhiệm vụ này không cần phân rã.

4.3.3. Phân tích nhiệm vụ phân cấp (6) – Ví dụ 2:

Biểu diễn phân tích công việc viết thư bằng ký pháp đồ họa:



4.3.4. Phân tích nhiệm vụ theo nhận thức (1)

- Phân tích nhiệm vụ theo nhận thức tập trung biểu diễn các tri thức mà người dùng có hoặc cần phải có để hoàn thành nhiệm vụ.
- Nhiệm vụ được định nghĩa là cái mà con người phải làm hay nghĩ là phải làm để hoàn thành mục đích.
- Để thực hiện nhiệm vụ, phải sử dụng một số thiết bị.
- Hai loại hành động:
 - Hành động vật lý: nhấn phím, di chuyển con trỏ, ...
 - Hành động có tính suy nghĩ: quyết định phím nào được nhấn, quyết định di chuyển con trỏ đến vị trí nào, ...

4.3.4. Phân tích nhiệm vụ theo nhận thức (2)

- Cơ sở: dựa vào lý thuyết nhận thức các hành động vật lý và hành động có tính suy nghĩ.
- Một số kỹ thuật phân tích nhiệm vụ nhận thức:
 - GOMS
 - TAG
 - ...

4.3.5. Phân tích nhiệm vụ theo mô hình tri thức (1)

- Còn gọi là mô hình “How to do”.
- Phân tích nhiệm vụ theo tri thức là liệt kê tất cả các đối tượng và hành động liên quan tới nhiệm vụ và sau đó phân loại chúng.
- Phân tích nhiệm vụ theo tri thức gần giống với HTA.
- Cơ sở: dựa vào ánh xạ nhiệm vụ – hành động. Nhiệm vụ là đích, đích con. Hành động là thao tác, phương pháp để thực hiện nhiệm vụ.
- GOMS là kỹ thuật thích hợp nhất để phân tích nhiệm vụ theo mô hình tri thức.

4.3.6. Sử dụng kỹ thuật phân tích nhiệm vụ (1)

- Thông tin nhận được từ quá trình phân tích nhiệm vụ phụ thuộc vào một số yếu tố:
 - Kỹ thuật sử dụng
 - Đối tượng áp dụng
 - Kế hoạch và chuỗi hành động sử dụng để thực hiện nhiệm vụ.
- Phân tích nhiệm vụ thường được sử dụng theo một trong ba cách:
 - Tạo tài liệu hướng dẫn
 - Tóm lược các yêu cầu và thiết kế hệ thống ở mức độ cao
 - Thiết kế giao tiếp chi tiết

4.3.6.1. Tạo tài liệu hướng dẫn (1)

- Nhằm hướng dẫn người dùng cách thực hiện một nhiệm vụ mới.
- Sử dụng kỹ thuật phân tích nhiệm vụ để hỗ trợ quá trình đào tạo.
- Có thể sử dụng cấu trúc phân cấp của HTA để cấu trúc tài liệu hoặc giáo trình.
- Tài liệu kiểu “How to do” thường hữu ích cho quá trình đào tạo ban đầu. Tuy nhiên, để cấu trúc một giáo trình hay thiết bị cho đào tạo cao hơn, tài liệu này phải được cấu trúc tốt hơn.
- Kỹ thuật phân tích nhiệm vụ có thể trợ giúp người dùng chuyển từ hệ thống này sang một hệ thống khác.

4.3.6.2. Tóm lược yêu cầu và thiết kế hệ thống ở mức độ cao (1)

- Kỹ thuật phân tích nhiệm vụ có thể được sử dụng để hướng dẫn thiết kế một hệ thống mới.
- Đối với các hệ thống mới, bản thân kỹ thuật phân tích nhiệm vụ không phải là một kỹ thuật để tóm lược các yêu cầu người dùng.
- Đối với các hệ thống đang tồn tại, kỹ thuật phân tích nhiệm vụ góp phần hoàn thiện các phát biểu về yêu cầu của người dùng để phát triển hệ thống.

4.3.6.2. Tóm lược yêu cầu và thiết kế hệ thống ở mức độ cao (2)

- Các phát biểu gốc do người dùng đưa ra thường liên quan đến:
 - Các thành phần mới được yêu cầu bổ sung cho hệ thống hiện tại.
 - Chức năng của hệ thống được phát triển.
- Thông tin liệt kê bởi người phát triển hệ thống thường tập trung miêu tả:
 - Cái mà hệ thống có thể làm (chức năng).
 - Không miêu tả cách sử dụng hệ thống.

4.3.6.2. Tóm lược yêu cầu và thiết kế hệ thống ở mức độ cao (3)

- Phân tích nhiệm vụ của hệ thống hiện tại có thể trợ giúp người phát triển hệ thống và người dùng theo hai cách:
 - Những đối tượng nào của hệ thống hiện tại có thể được đưa sang hệ thống mới.
 - Giúp người dùng hiểu rõ các đặc trưng mới thông qua các trình bày có khuôn dạng của các trạng thái công việc hiện tại.

4.3.6.3. *Thiết kế giao tiếp chi tiết*

- Kỹ thuật phân tích nhiệm vụ và đối tượng có thể sử dụng trong thiết kế giao tiếp chi tiết.
- Sử dụng cây phân cấp nhiệm vụ rất hữu ích cho quá trình này.
- Nhiệm vụ ở mức cao nhất được phân rã thành các nhiệm vụ con, các nhiệm vụ con này tiếp tục được phân rã, ...
- Chuỗi nhiệm vụ nhận được từ phân tích nhiệm vụ được dùng khi thiết kế đối thoại.
- Trình tự trong đó các nhiệm vụ con được thực hiện có thể sử dụng để so sánh với nhiệm vụ gốc.

Chương 4 – MÔ HÌNH HỆ THỐNG

4.1. Quy trình thiết kế hệ tương tác

4.2. Đặc tả yêu cầu người dùng

4.3. Phân tích nhiệm vụ

4.4. Mô hình thoại

Chương 4 – MÔ HÌNH HỆ THỐNG

4.4. Mô hình thoại

4.4.1. Khái niệm về đối thoại

4.4.2. Biểu diễn quá trình đối thoại dưới dạng sơ đồ

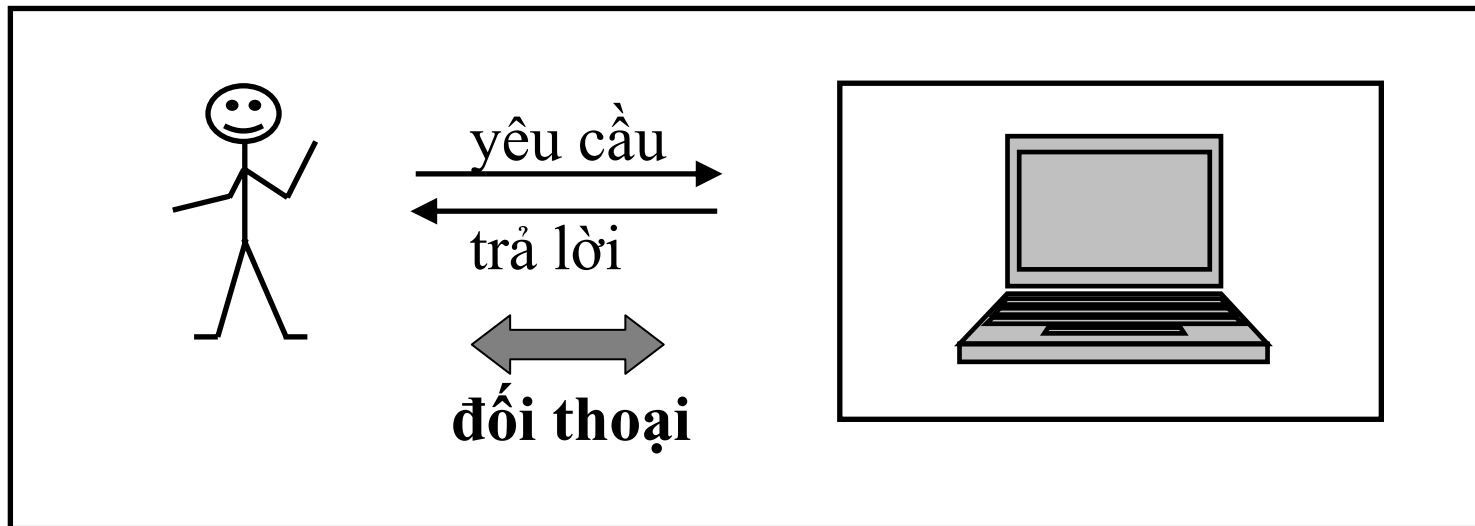
4.4.3. Biểu diễn quá trình đối thoại dưới dạng văn bản

4.4.4. Ngữ nghĩa thoại

4.4.5. Phân tích thiết kế thoại

4.4.1. Khái niệm về đối thoại (1)

- Đối thoại là mức độ ngữ nghĩa của HCI.
- Trong HCI, đối thoại là cuộc trao đổi giữa hai đối tác người dùng và máy tính.



4.4.1. Khái niệm về đối thoại (2)

- Ba mức độ của một ngôn ngữ máy tính:
 - Mức độ từ vựng (mức thấp nhất): hình dạng, biểu tượng, phím nhấn.
 - Mức độ cú pháp: thứ tự và cấu trúc của đầu vào, đầu ra.
 - Mức độ ngữ nghĩa: ý nghĩa của trao đổi (xét tính hiệu quả) trên cấu trúc dữ liệu bên trong của máy tính với thế giới bên ngoài.
- Trong giao tiếp với người sử dụng, đối thoại thường đồng nghĩa với mức độ cú pháp.

4.4.1. Khái niệm về đối thoại (3)

- Đối thoại người dùng có cấu trúc:
 - Đàm thoại người dùng (con người với con người):
 - + Thường không có cấu trúc.
 - + Các thành viên có thể trả lời các câu hỏi đã hoặc chưa xác định trước, tùy thuộc vào các tình huống khác nhau.
 - Đối thoại với máy tính (con người với máy tính):
 - + Thường có cấu trúc và bị ràng buộc.
 - + Đôi khi cũng không xác định hết các tình huống.
 - + Máy tính có thể đưa ra các thông báo khó lường.

4.4.1. *Khái niệm về đối thoại (4)*

- Ký pháp biểu diễn đối thoại:
 - Không phải ngôn ngữ lập trình.
 - Ngôn ngữ lập trình không đủ để biểu diễn đối thoại.
 - Cần sử dụng một ký pháp có khả năng tách riêng quá trình phân tích, tách riêng các phần tử giao tiếp của chương trình khỏi ngữ cảnh tính toán của nó.
 - Hai phương pháp dùng để mô tả đối thoại thường sử dụng:
 - + Ký pháp đồ họa (sơ đồ)
 - + Ký pháp văn bản

4.4.2. Biểu diễn quá trình đối thoại dưới dạng sơ đồ

- Được sử dụng phổ biến, do có tính trực quan.
- Sử dụng sơ đồ không phải lúc nào cũng tốt, nhất là đối với các đối thoại phức tạp.
- Các dạng ký pháp đồ họa tiêu biểu:
 - Mạng dịch chuyển trạng thái – STN
 - Mạng dịch chuyển trạng thái phân cấp – HSTN
 - Đối thoại tương tranh
 - Lưu đồ luồng – Flow Chart
 - Lưu đồ Jackson – JSD (Jackson Structured Design)

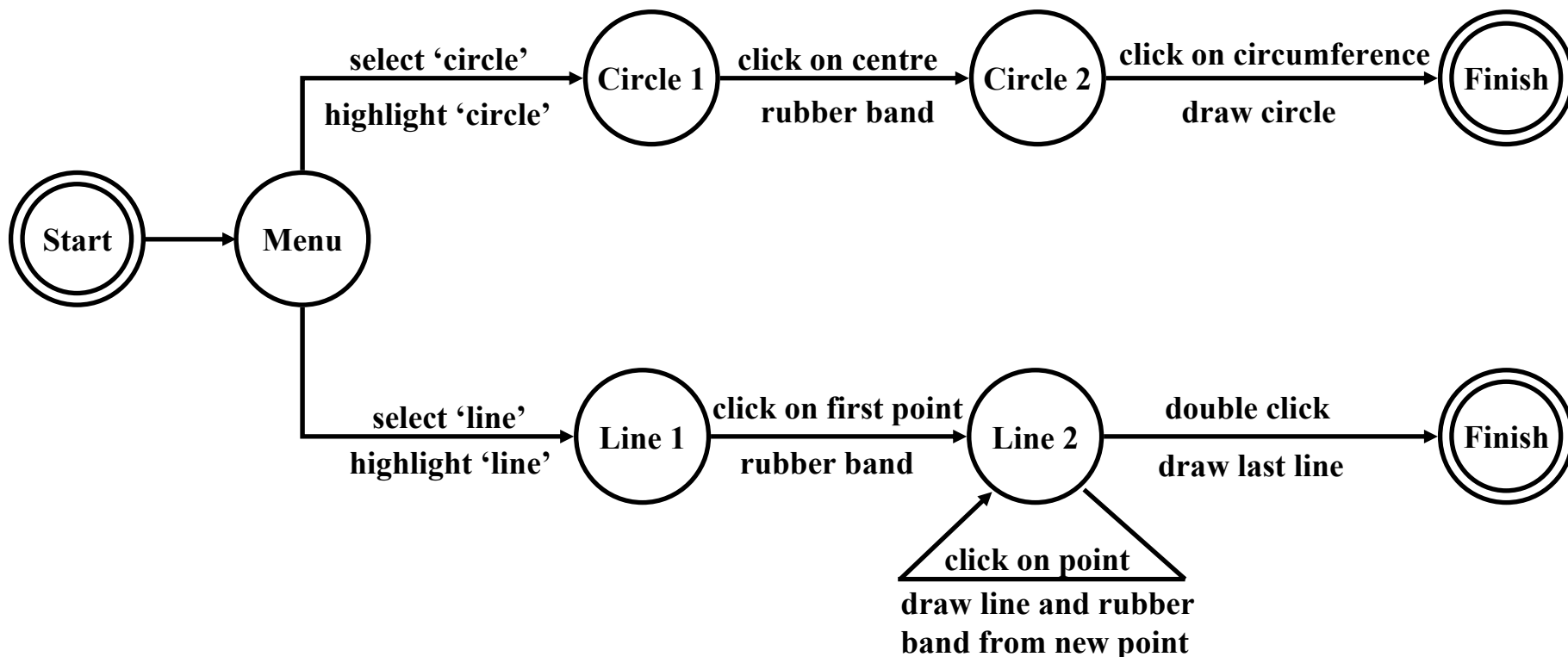
4.4.2.1. Mạng dịch chuyển trạng thái – STN (1)

- STN: State Transition Network.
 - Sử dụng hai đối tượng để mô tả đối thoại: vòng tròn và mũi tên.
 - Vòng tròn: mô tả một trạng thái của hệ thống.
 - Mũi tên: mô tả sự dịch chuyển trạng thái, hành động hay sự kiện.
- Trên mũi tên có thể có thông tin bổ sung.

4.4.2.1. Mạng dịch chuyển trạng thái (2) – Ví dụ:

- Mạng STN biểu diễn công cụ vẽ, sử dụng menu, gồm hai lựa chọn:
 - Vẽ đường tròn: từ hai điểm bất kỳ trên không gian vẽ
 - + Điểm thứ nhất: tâm đường tròn
 - + Điểm thứ hai: một điểm bất kỳ nằm trên đường tròn
 - Vẽ đoạn thẳng: từ hai điểm bất kỳ trên không gian vẽ (điểm đầu và điểm cuối của đoạn thẳng)

4.4.2.1. Mạng dịch chuyển trạng thái (3) – Ví dụ (tiếp):



4.4.2.2. Mạng dịch chuyển trạng thái phân cấp (1)

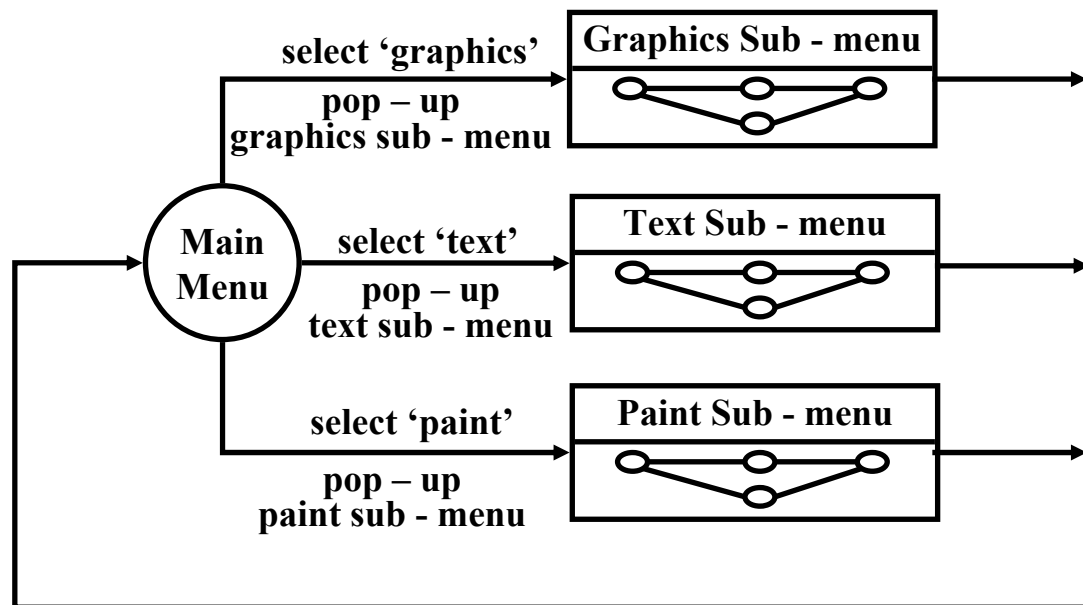
- Sử dụng khi đối thoại khá phức tạp.
- Đối thoại được chia thành các đối thoại nhỏ (sub – dialog).
- Để biểu diễn một trạng thái phức tạp, sử dụng thêm các mạng dịch chuyển trạng thái STN biểu diễn trong một hình chữ nhật.
- Nhờ sự phân cấp, các hệ thống lớn phức tạp được biểu diễn đơn giản hơn.

4.4.2.2. Mạng dịch chuyển trạng thái phân cấp (2) – Ví dụ:

- Hệ thống main menu có 3 lựa chọn (3 sub menu), được biểu diễn bởi:

- Menu đồ thị
- Menu văn bản
- Menu đồ họa

- Trong mỗi khối lại là các sub menu tương đương một trạng thái phức hợp.

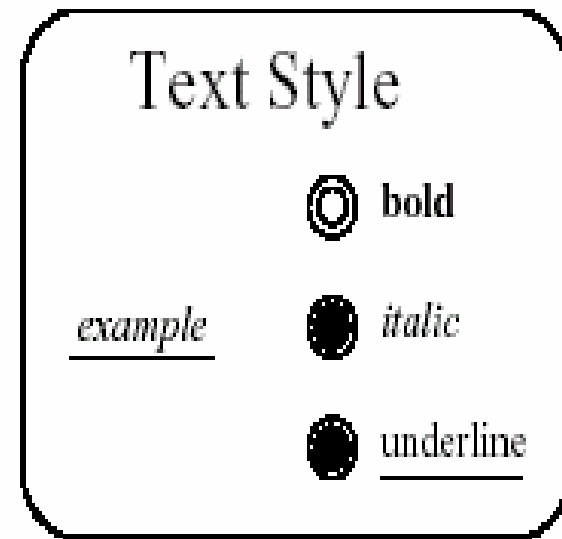


4.4.2.3. Đối thoại tương tranh (1)

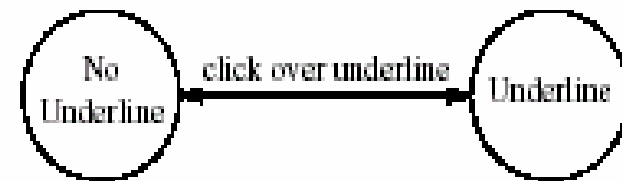
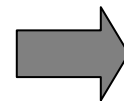
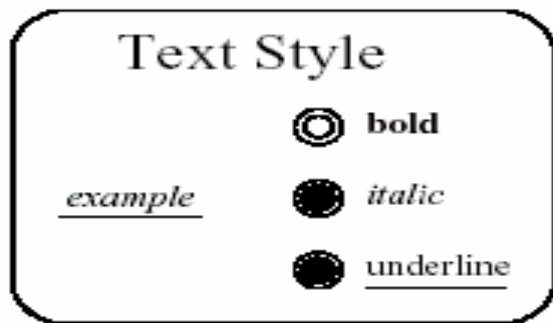
- Mạng dịch chuyển trạng thái STN rất phù hợp để biểu diễn đối thoại tuần tự, chọn hay lặp.
- STN rất khó để biểu diễn các đối thoại tương tranh.
- Tương tranh: là tính chất của một hệ thống bao gồm các công việc được thực thi trùng nhau về mặt thời gian.
- Để biểu diễn mỗi một công việc, có thể sử dụng một STN hai trạng thái.
- Để biểu diễn tổ hợp trạng thái, cần sử dụng tổ hợp lưu đồ.

4.4.2.3. Đối thoại tương tranh (2) – Ví dụ:

- Xét hộp hội thoại đơn giản kiểu văn bản:
 - Hộp lựa chọn chứa 3 phím Toggle. Mỗi phím cho một tình huống: đậm, nghiêng và gạch chân.
 - Một đoạn văn bản có thể là bất kỳ tổ hợp nào của 3 thuộc tính trên.

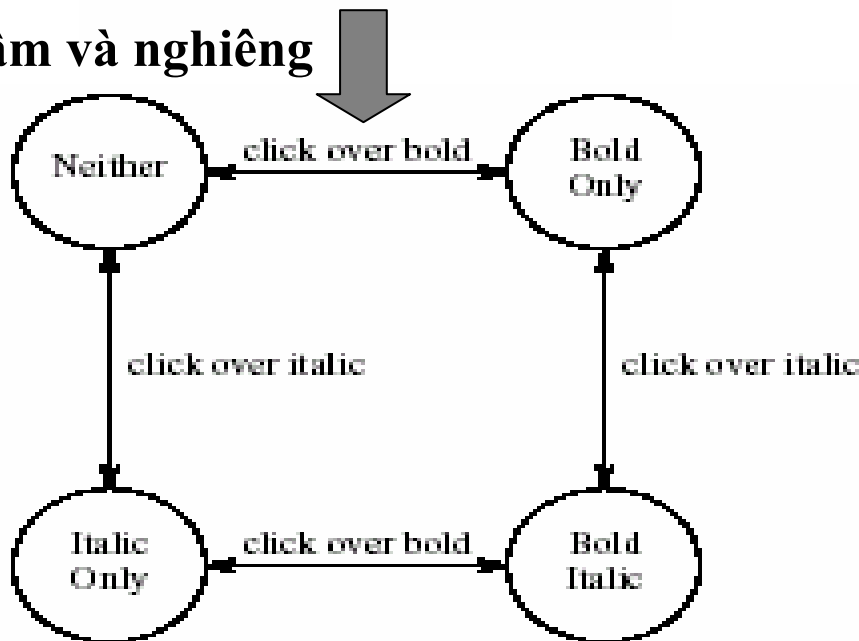


4.4.2.3. Đối thoại tương tranh (2) – Ví dụ (tiếp):



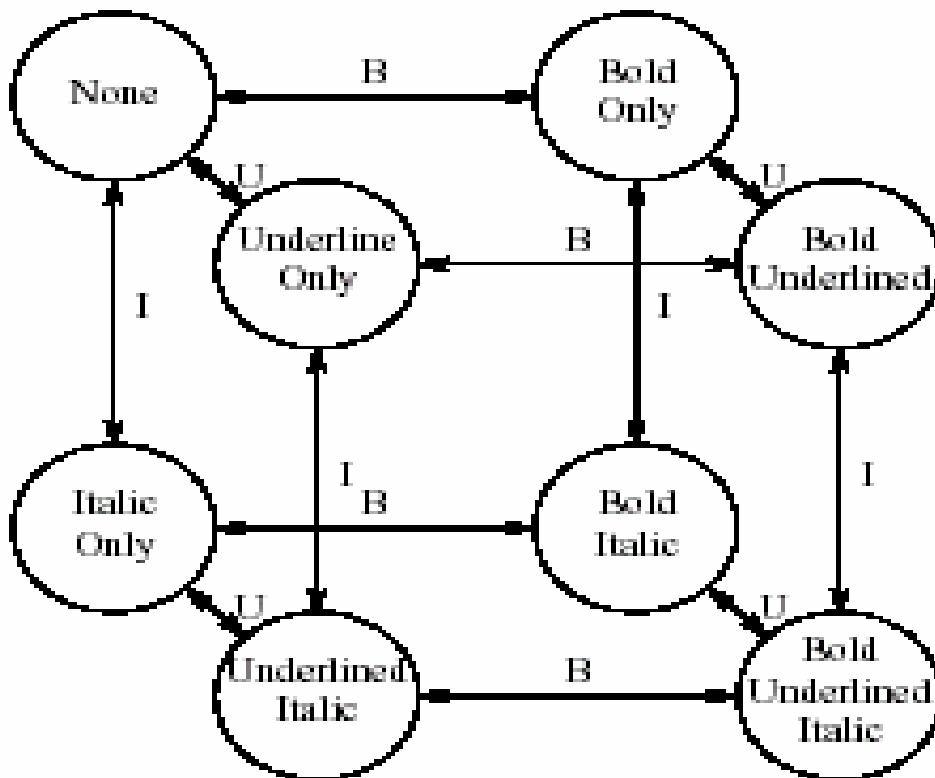
STN đơn cho mỗi phím

Tổ hợp đậm và nghiêng



4.4.2.3. Đối thoại tương tranh (3) – Ví dụ (tiếp):

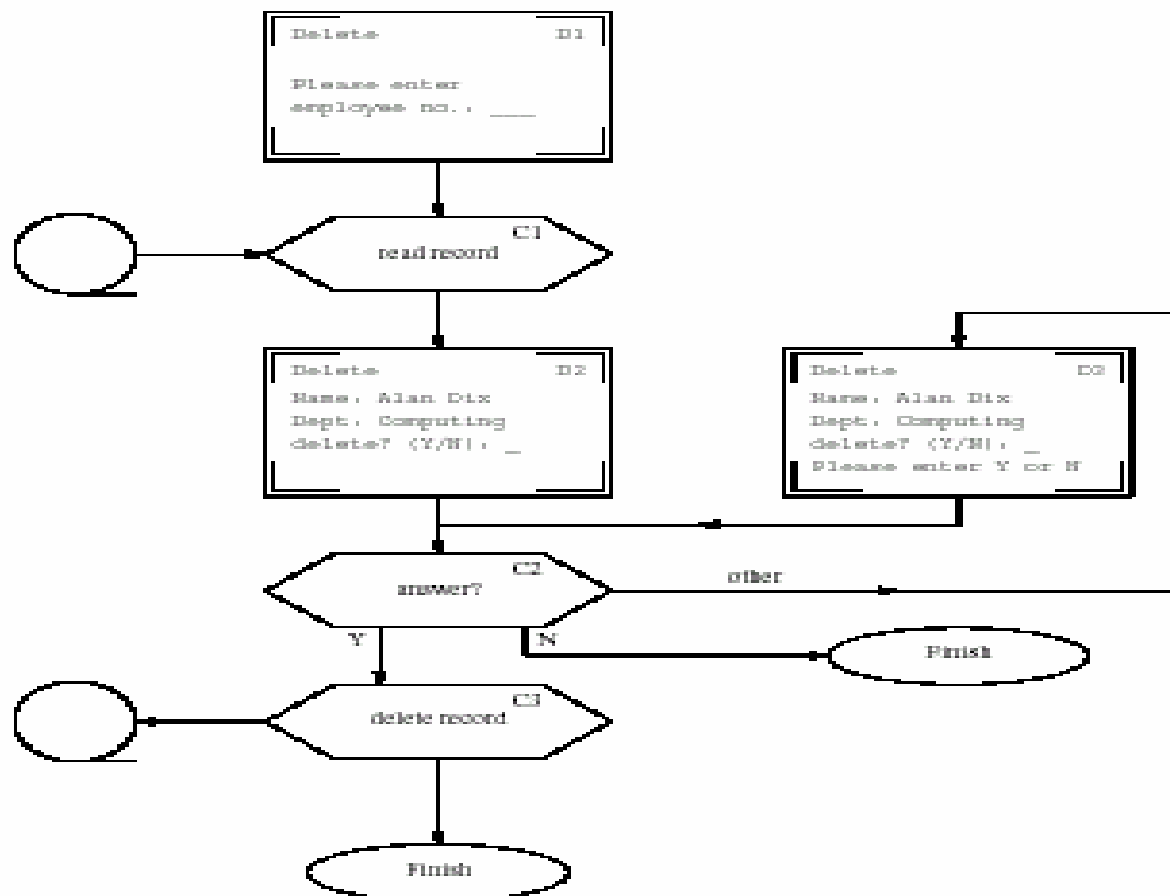
Số trạng thái có thể có của n toggle:



4.4.2.4. Lưu đồ luồng – Flow Chart (1)

- Đây là công cụ quen thuộc với lập trình viên.
- Sử dụng các thành phần: hộp hình chữ nhật, hộp đa giác, mũi tên, một số ký pháp vật lý khác.
 - Hộp: là quá trình hay một quyết định.
 - Sự kiện: không phải là trạng thái.
 - Sử dụng nhiều loại hộp khác nhau để biểu diễn các hoạt động khác nhau.

4.4.2.4. Lưu đồ luồng – Flow Chart (2) – Ví dụ:

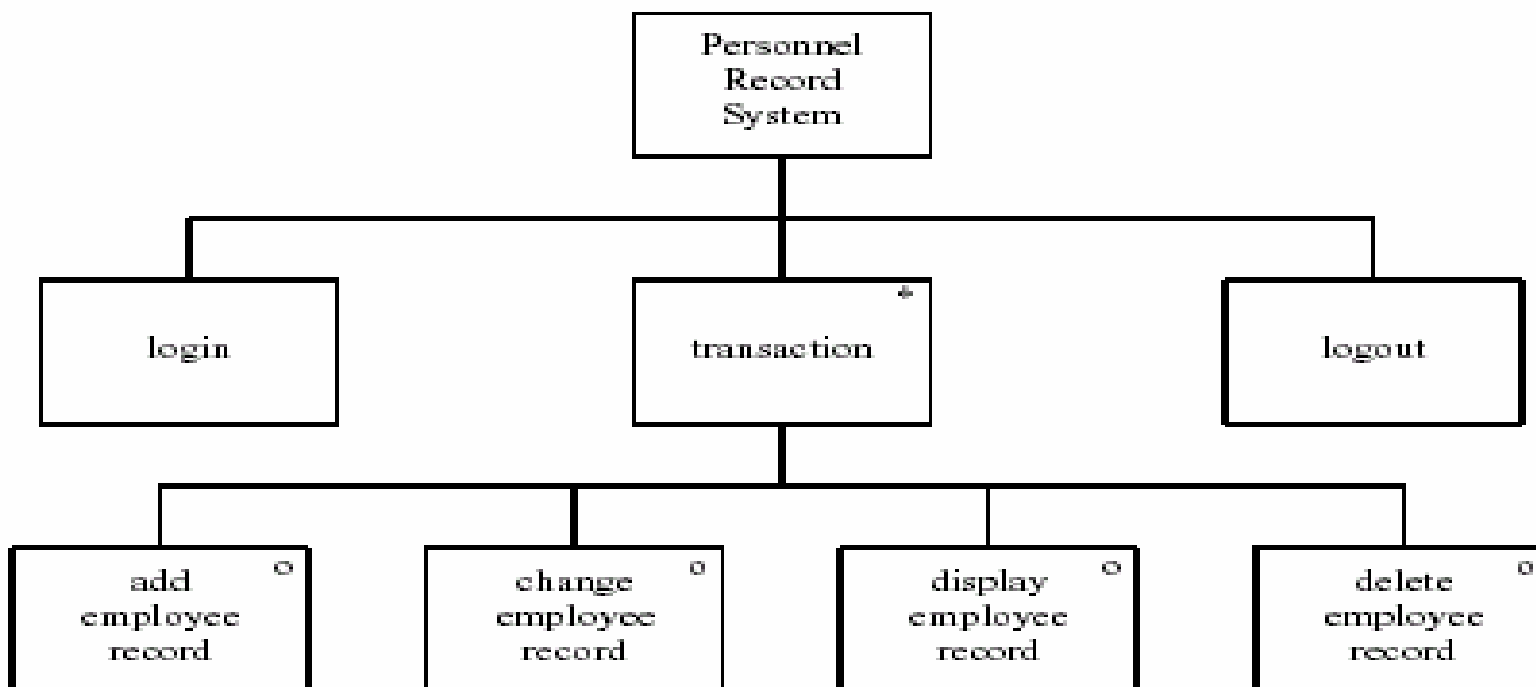


4.4.2.5. Lưu đồ Jackson – Jackson Structured Design (1)

- Ra đời sau lưu đồ luồng.
- Được sử dụng trong nhiều khía cạnh của phân tích nhiệm vụ TA.
- Lưu đồ JSD thích hợp với cấu trúc đối thoại hình cây.
- Sử dụng các hình hộp chữ nhật để biểu thị các hành động.
- Để biểu thị các hành động lặp đi lặp lại hoặc các hành động lựa chọn, sử dụng các dấu đặc biệt (như dấu ‘*’, dấu ‘o’).

4.4.2.5. Lưu đồ Jackson (2) – Ví dụ:

- Lưu đồ JSD diễn tả một hệ thống nhân sự, cho phép cập nhật thông tin về nhân sự: bổ sung, hiển thị, xóa,...



4.4.3. *Biểu diễn quá trình hội thoại dưới dạng văn bản*

- Ký pháp văn bản được sử dụng song song với ký pháp đồ họa.
- Một số loại ký pháp văn bản thường dùng:
 - Văn phạm
 - Luật sản xuất
 - CSP (Communicating Sequential Process) đại số

4.4.3.1. Văn phạm (1)

- Văn phạm được dùng khá phổ biến.
- Ký pháp BNF (Backus – Naur – Form) là một dạng hay được sử dụng trong mô tả ngôn ngữ lập trình và mô tả giao tiếp.
- Ký pháp văn phạm BNF đa dạng hơn so với biểu thức chính qui hay mạng dịch chuyển trạng thái STN.
- Tuy nhiên, ký pháp văn phạm không có khả năng biểu diễn tương tranh.

4.4.3.1. Văn phạm (2)

- Ký pháp chung của BNF:

$\langle \text{đối tượng} \rangle ::= \langle \text{giải nghĩa 1} \rangle | \langle \text{giải nghĩa 2} \rangle | \langle \text{giải nghĩa 3} \rangle | \dots$

- Ví dụ: ký pháp BNF cho định nghĩa một biểu thức chính qui:

$\text{expr} ::= \text{empty} | \text{atom expr} | \text{'('expr')'expr}$

Định nghĩa này được hiểu như sau:

biểu thức có thể là **rỗng** hoặc **biểu thức nguyên tố** hoặc **biểu thức**

4.4.3.2. Luật sản xuất (1)

- Luật sản xuất là khái niệm đã được nhắc đến trong mô hình Lý thuyết độ phức tạp nhận thức CCT.
- Là một chuỗi dạng: IF *điều kiện* THEN *hành động*.
- Các cách viết luật sản xuất:

cond -> action

cond: action

4.4.3.2. Luật sản xuất (2)

- Điều kiện: là một phát biểu về các nội dung của bộ nhớ làm việc.
- Hành động: có thể là dạng đáp ứng của người dùng hay sự thay đổi trạng thái của hệ thống.
- Nguyên tắc hoạt động của luật: khi các luật được kích hoạt, hệ thống so sánh phần *điều kiện* của luật với các sự kiện khởi tạo bởi người dùng. Khi *điều kiện* được thỏa mãn, *hành động* được thực hiện (gọi là luật “cháy”).
- Thứ tự xuất hiện của luật là không quan trọng.

4.4.3.2. Luật sản xuất (3)

- Hệ thống luật có thể là sự kiện, trạng thái hoặc cả sự kiện và trạng thái.
- Các loại sự kiện thường dùng:
 - Sự kiện người dùng: bắt đầu bằng chữ hoa.
 - Sự kiện trong: bắt đầu bằng chữ thường. Được dùng trong đối thoại để lưu lại vết của trạng thái đối thoại.
 - Đáp ứng của hệ thống: thường đặt trong ngoặc “<>”. Là hiệu ứng nhìn thấy hoặc nghe thấy của hệ thống.

4.4.3.2. Luật sản xuất (4)

- Một luật được gọi là “cháy” nếu mọi sự kiện trong phần điều kiện của luật đều hiện diện trong bộ nhớ, và mọi tương tác với người dùng được thực hiện ngay lập tức.
- Khi một luật “cháy”, mọi sự kiện trong điều kiện được loại khỏi hệ thống bộ nhớ, và các sự kiện trong phần hành động sẽ được bổ sung vào bộ nhớ.

4.4.3.2. Luật sản xuất (5) – Luật sản xuất hướng trạng thái

- Còn gọi là hệ thống luật mệnh đề PPS (Proposition Production System), do Olsen đề xuất.
- Hệ thống bộ nhớ cũng chứa một tập các giá trị tên. Tập tên này không bị loại khỏi bộ nhớ khi một luật “cháy”. Nó chỉ bị loại một cách tường minh bởi phần hành động của luật.
- Trạng thái của hệ thống được chia thành một tập hữu hạn các thuộc tính.
- Mỗi thuộc tính có thể nhận một tập giá trị.
- Các thuộc tính có thể được thiết lập như là kết quả hành động của người dùng, hoặc có tác động lên sự hiển thị của hệ thống.

4.4.3.3. CSP đại số (1)

- Ký pháp STN rất phù hợp cho mô tả tuần tự hay trạng thái. Ngược lại, luật sản xuất lại thích hợp cho mô tả tương tranh.
- Việc xử lý đồng thời (vừa tương tranh, vừa tuần tự) đặt ra trong nhiều bài toán.
- Quá trình đại số là hình thức được phát triển cho các quá trình bao gồm cả xử lý tương tranh và xử lý tuần tự.
- CSP (Communicating Sequential Process) được phát triển cho đặc tả đối thoại và đặc tả cho cả tuần tự và tương tranh.

4.4.3.3. CSP đại số (2)

- Một số quy ước:

Các sự kiện được đánh dấu “?”: hành động của người dùng

“=” : định nghĩa

-> : sự kiện tuần tự

“;” : quá trình tuần tự

[] : sự lựa chọn. Tên biểu diễn sự kiện

4.4.4. *Ngữ nghĩa thoại (1)*

- Nếu mô tả đối thoại chỉ nhằm mục đích thông báo giữa các nhà thiết kế: chỉ cần chú giải cho đối thoại hình thức với chủ ý của hành động.
- Nếu mô tả đối thoại dùng cho một đặc tả hình thức cho một hợp đồng hoặc để thực hiện một mẫu thử: phải miêu tả một cách hình thức đối thoại này bằng nhiều cách khác nhau.

4.4.4. *Ngữ nghĩa thoại (2)*

- Hai khía cạnh của ngữ nghĩa thoại:
 - Đối với ứng dụng (bên trong)
 - Đối với trình diễn (bên ngoài)
- Ba cách tiếp cận:
 - Ngữ nghĩa đặc tả ký pháp
 - Liên kết với ngôn ngữ lập trình
 - Liên kết với đặc tả hình thức

4.4.5. Phân tích thiết kế thoại (1)

- Các cách thức mà đối thoại có thể được phân tích nhằm phát hiện tính tiện dụng tiềm năng bằng cách xem xét các nguyên tắc.
- Các vấn đề cần quan tâm:
 - Hành động của người dùng
 - Trạng thái của đối thoại
 - Cách biểu diễn và từ vựng

4.4.5. Phân tích thiết kế thoại (2)

- Các tính chất của hành động:
 - Tính đầy đủ: cần xác định các cung bị thiếu.
 - Tính xác định:
 - + Có thể có nhiều cung cho một hành động
 - + Nếu có nhiều mức lồng nhau, phải thoát được
 - Tính nhất quán:
 - + Cùng một hành động, phải có cùng một hiệu quả
 - + Thể thức và tính quan sát được

4.4.5. Phân tích thiết kế thoại (3)

- Các tính chất của trạng thái:
 - Tính đạt tới được:
 - + Có thể nhận được mọi thứ từ bất kỳ vị trí nào
 - + Dễ dàng
 - Tính thuận nghịch:
 - + Có thể nhận được trạng thái trước?
 - + Nếu không, Undo
 - Các trạng thái nguy hiểm: một số trạng thái không mong muốn

Chương 5 – ĐÁNH GIÁ HỆ THỐNG

5.1. Vai trò đánh giá trong thiết kế tương tác

5.2. Các kiểu đánh giá

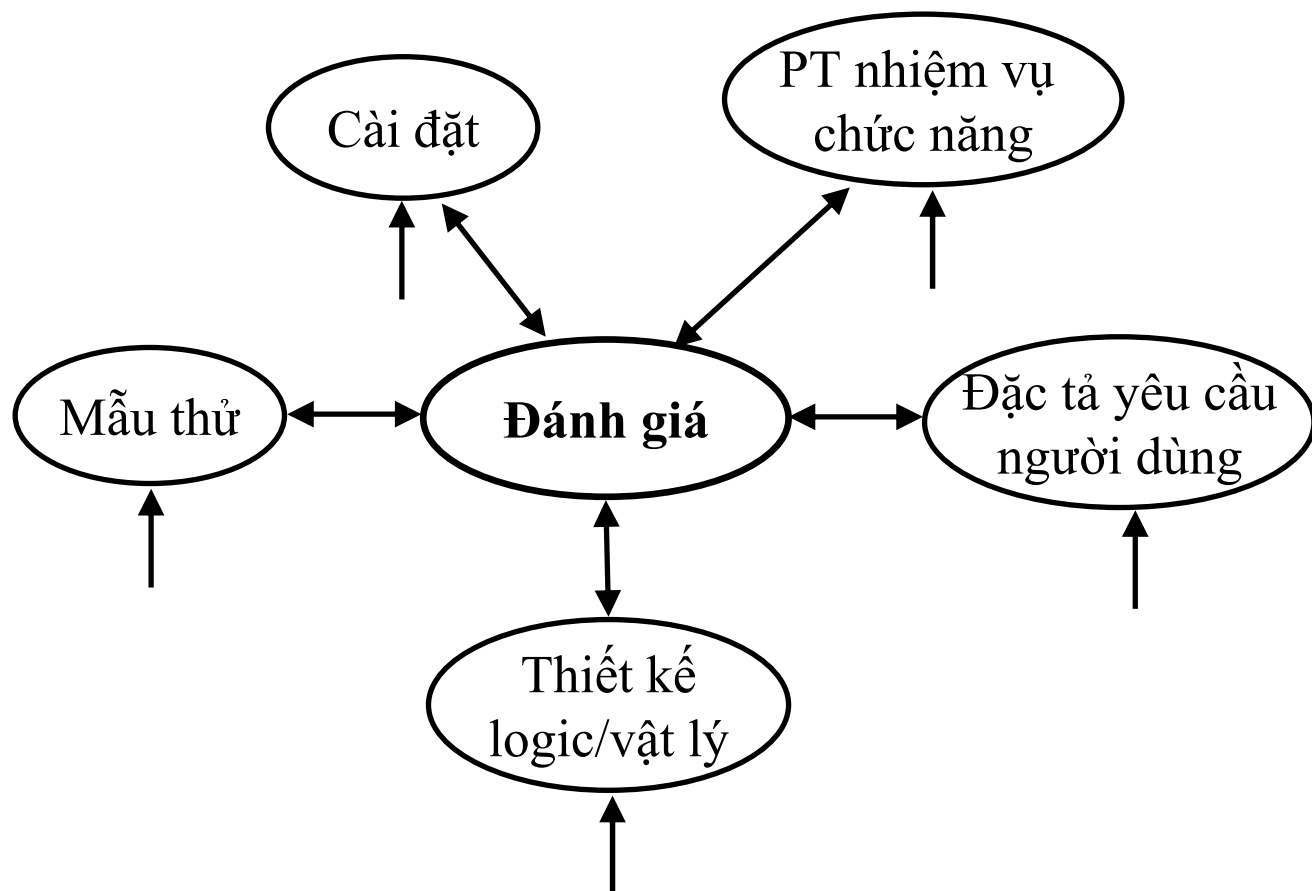
5.3. Đánh giá thiết kế

5.4. Đánh giá cài đặt

5.5. Lựa chọn phương pháp đánh giá

5.1. Vai trò đánh giá trong thiết kế tương tác (1)

- Quy trình thiết kế tương tác:
 - Đánh giá không phải là một giai đoạn trong quá trình thiết kế.
 - Đánh giá diễn ra trong suốt vòng đời của HCI.



5.1. Vai trò đánh giá trong thiết kế tương tác (2)

- Không thể thực hiện các kiểm thử thực nghiệm trong suốt quá trình thiết kế.
- Đánh giá: là quá trình thu thập dữ liệu kiểm tra về tính dùng được của thiết kế, đảm bảo ba nhiệm vụ chính:
 - Khẳng định tính mở rộng các chức năng của hệ thống.
 - Khẳng định tính hiệu quả của giao tiếp đối với người dùng.
 - Xác định một số vấn đề đặc biệt nảy sinh trong quá trình sử dụng.

5.1. Vai trò đánh giá trong thiết kế tương tác (3)

- Chức năng của hệ thống phải đáp ứng với đặc tả yêu cầu người dùng.
- Thiết kế hệ thống phải có khả năng đáp ứng các nhiệm vụ đặt ra một cách dễ dàng.
- Thiết kế hệ thống cũng bao hàm việc đánh giá khả năng sử dụng của hệ thống với những cái mà người dùng mong đợi.

5.1. Vai trò đánh giá trong thiết kế tương tác (4)

- Mục đích của đánh giá:
 - Đánh giá hệ thống về mặt chức năng.
 - Đánh giá sự ảnh hưởng của hệ thống đối với người dùng (tính dễ sử dụng, tính dễ học, ...).
 - Tải của hệ thống đối với người dùng (những kiến thức người dùng cần có khi sử dụng hệ thống, ...).

Chương 5 – ĐÁNH GIÁ HỆ THỐNG

5.1. Vai trò đánh giá trong thiết kế tương tác

5.2. Các kiểu đánh giá

5.3. Đánh giá thiết kế

5.4. Đánh giá cài đặt

5.5. Lựa chọn phương pháp đánh giá

5.2. Các kiểu đánh giá

- Phân loại theo điều kiện môi trường, nơi tiến hành đánh giá:
 - Đánh giá trong phòng thí nghiệm
 - Đánh giá thực địa
- Phân loại theo thời gian, vòng đời của quá trình thiết kế:
 - Đánh giá thiết kế
 - Đánh giá cài đặt

5.2.1. Đánh giá trong phòng thí nghiệm (1)

- Việc đánh giá diễn ra trong phòng thí nghiệm.
- Thường được sử dụng trong quá trình thiết kế.
- Các bên tham gia:
 - Người đánh giá.
 - Người dùng (có thể có hoặc không tham gia).
- Mục đích:
 - Người đánh giá muốn thực hiện một số khẳng định của mình.
 - Không nhất thiết người dùng phải có mặt.

5.2.1. Đánh giá trong phòng thí nghiệm (2)

- Ưu điểm:
 - Tiến hành trong phòng thí nghiệm nên có điều kiện hơn.
 - Một số phòng thí nghiệm được trang bị các thiết bị để nghiên cứu tính dùng được: thiết bị nghe nhìn, camera, gương phản chiếu, ... (không có trên thực địa).
- Nhược điểm:
 - Thiếu ngữ cảnh, môi trường không tự nhiên, không có thật.
 - Gặp khó khăn khi phải quan sát nhiều người cùng thực hiện một nhiệm vụ trong phòng thí nghiệm, do việc giao tiếp giữa con người với nhau phụ thuộc nhiều vào ngữ cảnh.

5.2.1. *Đánh giá trong phòng thí nghiệm (3)*

- Ứng dụng:
 - Khi môi trường thực địa không cho phép (ví dụ: môi trường nguy hiểm, môi trường ở xa,...)
 - Khi muốn phát hiện một số vấn đề hay một số thủ tục ít dùng.
 - Khi muốn so sánh các thiết kế khác nhau.

5.2.2. *Đánh giá thực địa (1)*

- Việc đánh giá thực hiện trong môi trường người dùng.
- Có thể diễn ra trong giai đoạn thiết kế hay giai đoạn cài đặt.
- Các bên tham gia:
 - Người đánh giá.
 - Người dùng (bắt buộc phải có sự tham gia của người dùng).
- Mục đích:
 - Đánh giá hệ thống trong hoạt động.
 - Đánh giá trạng thái người dùng.

5.2.2. *Đánh giá thực địa (2)*

- Ưu điểm:
 - Ngữ cảnh là ngữ cảnh thật.
 - Người dùng làm việc trong môi trường thật.
 - Quan sát được sự tương tác giữa hệ thống, giữa các cá nhân (không thể có khi tiến hành trong phòng thí nghiệm).
- Nhược điểm: đánh giá có thể có thiếu sót, do ảnh hưởng của nhiều yếu tố (tiếng ồn, người chuyển động, ...) gây mất tập trung.

5.2.2. *Đánh giá thực địa (3)*

- Ứng dụng:
 - Khi cần nghiên cứu sự tương tác và những gì xảy ra trong ngữ cảnh thật.
 - Khi cần nghiên cứu các trạng thái đặc biệt (như khi lưu trữ hay khôi phục một nhiệm vụ).

Chương 5 – ĐÁNH GIÁ HỆ THỐNG

5.1. Vai trò đánh giá trong thiết kế tương tác

5.2. Các kiểu đánh giá

5.3. Đánh giá thiết kế

5.4. Đánh giá cài đặt

5.5. Lựa chọn phương pháp đánh giá

Chương 5 – ĐÁNH GIÁ HỆ THỐNG

5.3. Đánh giá thiết kế

5.3.1. *Khái niệm*

5.3.2. *Mô hình Cognitive Walkthrough*

5.3.3. *Đánh giá Heuristic*

5.3.4. *Đánh giá nhờ xem xét (review based)*

5.3.5. *Đánh giá sử dụng mô hình*

5.3.1. *Khái niệm (1)*

- Đánh giá thiết kế: việc đánh giá diễn ra ngay trong quá trình thiết kế.
- Đánh giá nên được thực hiện trước khi hệ thống được cài đặt.
- Lỗi được phát hiện sớm (ngay trong quá trình thiết kế) sẽ tránh được những ảnh hưởng đáng tiếc.
- Phát hiện lỗi muộn thì chi phí chỉnh sửa sẽ cao hơn.
- Phần lớn không cần sự có mặt của người dùng.
- Mục đích: xác định các vấn đề có thể là nguyên nhân gây khó khăn khi thực hiện hệ thống.

5.3.1. *Khái niệm (2)*

- Các kỹ thuật thường sử dụng khi đánh giá thiết kế:
 - Mô hình Cognitive WalkThrough
 - Đánh giá kiểu Heuristic
 - Đánh giá nhờ xem xét (review based)
 - Đánh giá sử dụng mô hình

5.3.2. Mô hình *Cognitive WalkThrough* (1)

- Được đề xuất bởi Polson và các đồng nghiệp.
- Sử dụng lý thuyết tâm lý học nhận thức vào kỹ thuật Walkthrough (kỹ thuật kiểm tra từ đầu đến cuối).
- Là phương pháp có tính dự đoán, nhằm phát hiện vấn đề từ rất sớm.

5.3.2. Mô hình *Cognitive WalkThrough* (2)

- Walkthrough yêu cầu xem xét một cách chi tiết chuỗi các hành động.
- Người đánh giá thực hiện theo các bước định nghĩa từ các nhiệm vụ đặc tả hệ thống, nhằm kiểm tra tính dùng được của hệ thống.
- Kỹ thuật này nhằm thiết lập tính dễ học và dễ dùng của hệ thống thông qua cách học theo kiểu “khám phá”.
- Đưa ra các câu hỏi liên quan đến việc khám phá.

5.3.2. Mô hình *Cognitive WalkThrough* (3)

- Các thông tin cần thiết cho kỹ thuật *Cognitive WalkThrough*:
 - Mô tả nguyên mẫu của hệ thống. Mô tả này không cần đầy đủ, song cũng nên khá chi tiết.
 - Mô tả về nhiệm vụ mà người dùng phải thực hiện. Nhiệm vụ phải mang tính tiêu biểu, là cái mà người dùng hay làm.
 - Danh sách chi tiết các hành động cần thiết để hoàn thành nhiệm vụ theo nguyên mẫu.
 - Chỉ dẫn về người dùng là ai, các tri thức, kinh nghiệm mà người đánh giá có thể giả định.
- Với thông tin trên, người đánh giá thực hiện chuỗi các hành động và đưa ra các ý kiến về tính dùng được.

5.3.2. Mô hình *Cognitive WalkThrough* (4)

- Với mỗi hành động, người đánh giá trả lời các câu hỏi:
 - Người dùng có tạo ra bất cứ tác động gì trên hành động đó không?
 - Người dùng có khả năng để ý hành động đúng là có?
 - Khi người dùng tìm thấy một hành động đúng trên giao tiếp, liệu họ có biết đó là cái duy nhất đúng cho mục đích mà họ cố tạo ra?
 - Sau khi hành động tiến hành, người dùng sẽ hiểu phản hồi của hệ thống? Giả sử người dùng đã chọn đúng hành động, họ có biết hành động tiếp theo là gì hay không?

5.3.2. Mô hình *Cognitive WalkThrough* (5)

- Bảng thực hiện đánh giá nhận thức Walkthrough:

Giao diện:

Nhiệm vụ:

Người đánh giá: Ngày tháng:.....

Mô tả nhiệm vụ:

Chuỗi hành động:

Đoán trước của người dùng:

Đích ban đầu của người dùng:

Hành động tiếp theo:Mô tả:

5.3.2. Mô hình *Cognitive WalkThrough* (6)

- Bảng thực hiện đánh giá nhận thức Walkthroug (tiếp):
 1. Hiệu chỉnh mục tiêu (đích):
 -
 -
 3. Vấn đề xác định hành động:
 -
 -
 4. Vấn đề thực hiện hành động:
 -
 -

5.3.3. *Đánh giá Heuristic (1)*

- Là các nguyên tắc chung, các hướng dẫn, hay các luật có thể:
 - Trợ giúp một quyết định trong thiết kế.
 - Phê phán một quyết định đã được thực hiện.
- Được đề xuất bởi Jacob Nielsen và Rolf Molich.
- Được sử dụng trong giai đoạn đầu của thiết kế, nhằm cố định tính dùng được.
- Nhiều người đánh giá độc lập cùng tiến hành trên một hệ thống, để nêu lên tính dùng được.

5.3.3. *Đánh giá Heuristic (2) – 9 heuristics*

1. Đối thoại đơn giản và tự nhiên:

- Đơn giản: sử dụng ít thông tin.
- Tự nhiên: lệnh gần với yêu cầu.

2. Sử dụng ngôn ngữ của người dùng:

- Sử dụng cách nói của người dùng.
- Không dùng thuật ngữ công nghệ đặc biệt.

3. Giảm tải tối thiểu cho bộ nhớ người dùng:

- Không bắt người dùng phải nhớ cho hành động tiếp sau.
- Lưu lại thông tin trên màn hình cho đến khi người dùng không cần nữa.

5.3.3. Đánh giá Heuristic (3) – 9 heuristics

4. Tính nhất quán:

- Dãy các hành động phải học trong phần này phải áp dụng được cho các phần khác.

5. Phản hồi thông tin:

- Người dùng nhận biết được tác động của hành động mình đã thực hiện trên hệ thống.

6. Xử lý tình huống thoát:

- Khi người dùng gặp một tình huống/vấn đề mà họ không quan tâm, họ có thể thoát ra một cách nhanh chóng và không làm tổn hại đến bất cứ một cái nào khác.

5.3.3. *Đánh giá Heuristic (4) – 9 heuristics*

7. Tạo lỗi tắt:

- Giúp đỡ người dùng có kinh nghiệm tránh những đối thoại và thông tin không cần thiết.

8. Có thông báo lỗi tốt:

- Giúp người dùng hiểu nguyên nhân của sự việc và có cách để khắc phục.

9. Dự báo lỗi:

5.3.3. *Đánh giá Heuristic (5) – 10 quy định*

1. Thấy được các trạng thái của hệ thống: hệ thống luôn cho người dùng thấy cái họ sẽ làm qua thông tin phản hồi trong thời gian hợp lý.
2. Sự tự do và quản lý của người dùng: người dùng có thể gặp lỗi, do đó phải có thoát khẩn cấp.
3. Dự báo lỗi.
4. Chuẩn hóa và nhất quán.
5. Đối sánh hệ thống với thế giới thực.

5.3.3. *Đánh giá Heuristic (6) – 10 quy định*

6. Nhận biết chứ không phải nhớ lại: tạo cho đối tượng, hành động và các lựa chọn dễ dàng nhận biết bởi người dùng chứ không cần nhớ.
7. Sử dụng hiệu quả và mềm dẻo: dùng được cho cả hai loại người dùng có và không có kinh nghiệm.
8. Thiết kế đơn giản và có thẩm mỹ.
9. Trợ giúp người dùng nhận biết, đối thoại và phục hồi từ trạng thái lỗi.
10. Trợ giúp và tài liệu.

5.3.3. *Đánh giá Heuristic (7) – Thực hiện đánh giá*

- Cần phải điều chỉnh mọi vấn đề trong danh sách, diễn giải tại sao nó vi phạm heuristic.
- Liệt kê từng vấn đề mà người đánh giá phát hiện ra.
- Kiểm tra giao diện ít nhất hai lần.

5.3.3. Đánh giá Heuristic (8) – Tiến trình đánh giá

- Gồm 4 bước:
 - Thông báo
 - Đánh giá
 - Xếp hạng lỗi nghiêm trọng
 - Bàn bạc

5.3.3. *Đánh giá Heuristic (8) – Tiến trình đánh giá*

- Thông báo:
 - Tổ chức gặp gỡ giữa đội ngũ thiết kế và những người đánh giá.
 - Giới thiệu ứng dụng.
 - Giải thích về số đông người sử dụng, nghiệp vụ và các kịch bản.

5.3.3. *Đánh giá Heuristic (9) – Tiến trình đánh giá*

- **Đánh giá:**
 - Người đánh giá làm việc độc lập.
 - Người quan sát viết báo cáo hay ghi âm lại những bình luận của người đánh giá.
 - Người đánh giá tập trung vào phát hiện vấn đề, không xếp hạng tính nguy hại của vấn đề tại thời điểm này.

5.3.3. *Đánh giá Heuristic (10) – Tiến trình đánh giá*

- Xếp hạng lỗi nghiêm trọng:
 - Người đánh giá xếp mức ưu tiên cho mọi vấn đề tìm ra (của cả nhóm người đánh giá).
 - Giải thích ý nghĩa cách xếp hạng.
- Bàn bạc:
 - Người đánh giá và đội ngũ thiết kế trao đổi các kết quả và phương pháp giải quyết vấn đề.

5.3.4. Đánh giá nhờ xem xét – review based

- Là đánh giá kiểu Heuristic.
- Được Molich và Nelson đưa ra năm 1990.
- Sử dụng trong các ứng dụng nhỏ.
- Thời gian đánh giá thấp (khoảng 2 giờ). Nếu ứng dụng lớn, thời gian có thể nhiều hơn.
- Chi phí đánh giá thấp, tài nguyên sử dụng không nhiều.

5.3.5. *Đánh giá sử dụng mô hình*

- Tổ hợp đặc tả thiết kế và đánh giá trong cùng một khung tương tác.
- Là kỹ thuật đánh giá đòi hỏi:
 - Đặc tả chức năng của hệ thống có liên quan.
 - Phân tích nhiệm vụ chứa danh sách nhiệm vụ và gán thành các thành phần.
 - Cấu trúc nhiệm vụ từ đơn giản đến phức tạp.
 - Các thao tác người dùng có thể đánh giá bằng phương pháp giải tích.
- Thường áp dụng mô hình vật lý Keystrock.

Chương 5 – ĐÁNH GIÁ HỆ THỐNG

5.1. Vai trò đánh giá trong thiết kế tương tác

5.2. Các kiểu đánh giá

5.3. Đánh giá thiết kế

5.4. Đánh giá cài đặt

5.5. Lựa chọn phương pháp đánh giá

Chương 5 – ĐÁNH GIÁ HỆ THỐNG

5.4. Đánh giá cài đặt

5.4.1. *Khái niệm*

5.4.2. *Đánh giá thực nghiệm*

5.4.3. *Kỹ thuật quan sát*

5.4.4. *Kỹ thuật hỏi đáp*

5.4.1. *Khái niệm*

- Đánh giá cài đặt: là kỹ thuật đánh giá có sự hiện diện của người dùng và hệ thống đã được cài đặt.
- Có thể sử dụng mô phỏng hay mẫu thử.
- Ba kỹ thuật đánh giá cài đặt thường sử dụng:
 - Đánh giá thực nghiệm.
 - Kỹ thuật quan sát.
 - Kỹ thuật hỏi đáp.

5.4.2. Đánh giá thực nghiệm (1)

- Mục đích: đánh giá thiết kế hoặc một khía cạnh nào đó của thiết kế.
- Cung cấp một chứng cứ thực nghiệm để hỗ trợ một yêu cầu hoặc một giả thiết cụ thể.
- Cách thức tiến hành: người đánh giá lựa chọn giả thiết đánh giá thông qua một số thuộc tính của chủ thể nhằm đánh giá một số khía cạnh đặc trưng của thiết kế.
- Ba thông số quan trọng trong kỹ thuật đánh giá thực nghiệm: chủ đề, biến và giả thiết.

5.4.2. *Đánh giá thực nghiệm (2) – Chủ đề*

- Việc lựa chọn chủ đề quyết định tới thành công của đánh giá.
- Chủ đề chọn phải tương ứng với kỳ vọng của lớp người dùng, chủ đề càng gần với kỳ vọng càng tốt.
- Lý tưởng: đánh giá với sự hiện diện của người dùng.
- Nếu người dùng không phải là người dùng thực sự thì phải chọn lớp tương tự: cùng độ tuổi, tri thức, ...
- Kích thước tập mẫu của chủ đề phải đủ lớn và đặc trưng (ít nhất 10 mẫu).

5.4.2. *Đánh giá thực nghiệm (3) – Biến/đại lượng*

- Các đại lượng được quản lý và đo đạc dưới điều kiện thực nghiệm nhằm kiểm tra giả thiết.
- Có hai loại biến (đại lượng):
 - Loại biến quản lý (biến độc lập)
 - Loại biến đo lường (biến phụ thuộc).

5.4.2. *Đánh giá thực nghiệm (4) – Biến/đại lượng*

- Biến độc lập:
 - Là các đặc trưng của thực nghiệm.
 - Tạo ra các điều kiện so sánh khác nhau, như: kiểu giao tiếp, mức độ trợ giúp, số mức menu, các icon, ...
 - Các kiểm tra phức tạp cần số biến độc lập càng nhiều.
 - Mỗi biến có thể cung cấp các giá trị khác nhau, gọi là mức độ của biến.

5.4.2. *Đánh giá thực nghiệm (5) – Biến/đại lượng*

- Biến phụ thuộc:
 - Là các biến có thể đo đếm được theo nhiều cách.
 - Ví dụ: trong giao tiếp dùng menu, biến phụ thuộc có thể là tốc độ lựa chọn menu.
 - Biến phụ thuộc thường được chọn để biểu diễn các độ đo (như thời gian thực hiện nhiệm vụ, số lỗi mắc phải, ...)
 - Số lượng biến phụ thuộc càng nhiều càng tốt (để so sánh với các số đo định trước).

5.4.2. *Đánh giá thực nghiệm (6) – Giả thiết*

- Giả thiết: là một số dự đoán kết quả của thực nghiệm được hình thành từ biến độc lập và biến phụ thuộc.
- Mục đích của thực nghiệm: chứng minh các dự đoán là chính xác.
- Thường dùng các số đo thống kê để so sánh với các mức độ có nghĩa khác nhau.

5.4.2. *Đánh giá thực nghiệm (7) – Thiết kế thực nghiệm*

- Thiết kế đánh giá thực nghiệm để tạo ra kết quả có độ tin cậy và có tính khái quát.
- Các yếu tố phải xem xét khi thiết kế thực nghiệm:
 - Chủ đề
 - Biến độc lập và biến phụ thuộc
 - Giả thiết

5.4.2. *Đánh giá thực nghiệm (8) – Thiết kế thực nghiệm*

- Thiết kế đánh giá thực nghiệm gồm hai bước:
 - Xem xét các yếu tố mà thực nghiệm phải coi như chủ đề, biến và giả thiết để quyết định chính xác cái mà người đánh giá đang thử chứng minh.
 - Lựa chọn phương pháp sử dụng: giữa các nhóm và trong nội bộ nhóm.

5.4.2. *Đánh giá thực nghiệm (9) – Thiết kế thực nghiệm*

- Phương pháp giữa các nhóm:
 - Mỗi một chủ đề được gán cho một điều kiện khác nhau.
 - Phải có ít nhất hai điều kiện: điều kiện thực nghiệm (trong đó các biến được điều khiển) và điều kiện kiểm tra.
 - Kiểm tra nhằm khẳng định điều khiển sẽ chịu trách nhiệm về những cái khác nhau sẽ được đo đếm.
 - Số lượng nhóm phụ thuộc vào số lượng biến độc lập (thường chọn lớn hơn hai nhóm).

5.4.2. *Đánh giá thực nghiệm (10) – Thiết kế thực nghiệm*

- Phương pháp giữa các nhóm (tiếp):
 - Ưu điểm:
 - + Mọi tác động của việc học thu được từ kết quả thực hiện của người dùng trong một điều kiện sẽ được kiểm tra.
 - + Mỗi người dùng chỉ thực hiện trong một điều kiện.
 - Nhược điểm:
 - + Đòi hỏi nhiều chủ đề.
 - + Cần chọn lựa chủ đề cẩn thận để đảm bảo đại diện cho tất cả người dùng.

5.4.2. *Đánh giá thực nghiệm (11) – Thiết kế thực nghiệm*

- Phương pháp trong nhóm:
 - Mỗi người dùng sẽ thực hiện trong điều kiện khác nhau.
 - Ưu điểm:
 - + Chi phí thấp, do cần ít người dùng.
 - + Nếu có đào tạo, sẽ đạt hiệu quả cao.

5.4.2. *Đánh giá thực nghiệm (12) – Phân tích kết quả*

- Thường sử dụng phương pháp phân tích thống kê.
- Hai luật cho phân tích thống kê: Quan sát và Lưu trữ dữ liệu.
- Quan sát: thực hiện kiểm tra thống kê qua đồ thị, lưu đồ hoặc bảng giá trị.
- Lưu trữ: rất cần cho các phương pháp phân tích về sau.

5.4.3. Kỹ thuật quan sát (1)

- Quan sát cách người dùng tương tác với hệ thống để thu thập thông tin về việc sử dụng hiện thời của hệ thống.
- Người dùng thường được hỏi về việc thực hiện một tập các nhiệm vụ định trước.
- Việc quan sát được tiến hành với sự hiện diện của người dùng.
- Có thể quan sát hoặc dùng video.
- Hai kỹ thuật thường dùng: quan sát trực tiếp, quan sát gián tiếp.

5.4.3. Kỹ thuật quan sát (2)

- Quan sát trực tiếp:
 - Quan sát người dùng ở trạng thái đang làm việc.
 - Hiệu quả cao.
 - Giảm trung gian.
 - Người dùng có thể mất tập trung, do có sự hiện diện của người đánh giá.

5.4.3. Kỹ thuật quan sát (3)

- Quan sát gián tiếp:
 - Quay video.
 - Phân tích kết quả thu nhận được.
 - Sử dụng hai kỹ thuật: “Think aloud” và đánh giá tập thể.

5.4.3. Kỹ thuật quan sát (4) – Kỹ thuật “Think aloud”

- Mô tả cái người dùng tin sẽ xảy ra, tại sao lại hành động như vậy, người dùng sẽ cố làm việc gì.
- Ưu điểm: đơn giản, đòi hỏi ít tri thức để thực hiện, cung cấp sự hiểu biết thấu đáo về các vấn đề với một giao tiếp.
- Có thể ứng dụng trong giai đoạn đầu của thiết kế.
- Quá trình quan sát có thể ảnh hưởng đến cách thức mà người dùng thực hiện nhiệm vụ.

5.4.3. Kỹ thuật quan sát (5) – Kỹ thuật đánh giá tập thể

- Là một biến thể của “Think aloud”.
- Người dùng cố gắng xem mình như một thành viên trong quá trình đánh giá, chứ không phải là chủ thể bị thực nghiệm.
- Người đánh giá có thể hỏi người dùng các câu hỏi dạng: tại sao, cái gì, nếu ...

5.4.3. Kỹ thuật quan sát (6) – Kỹ thuật đánh giá tập thể

- Ưu điểm:
 - Quy trình ít bị ràng buộc, dễ học bởi người đánh giá.
 - Người dùng cố gắng phê phán hệ thống.
 - Người đánh giá có thể chỉ rõ điểm nhầm lẫn ở thời điểm xảy ra và do đó có thể đạt hiệu quả tối đa.

5.4.3. Kỹ thuật quan sát (7) – Phân tích kết quả

- Là các cách ghi lại hành động người dùng.
- Một số phương pháp thường dùng:
 - Bút và giấy.
 - Ghi âm.
 - Ghi hình.
 - Nhật ký máy tính.
 - Sử dụng công cụ phân tích thể thức tự động.

5.4.4. Kỹ thuật hỏi đáp (1)

- Ít trực quan hơn so với các kỹ thuật thực nghiệm.
- Rất có ích trong việc liệt kê chi tiết quan điểm của người dùng về hệ thống.
- Có thể dùng trong đánh giá và trong sưu tập thông tin về yêu cầu của người dùng và nhiệm vụ.

5.4.4. Kỹ thuật hỏi đáp (2)

- Ưu điểm:
 - Có thể nhận được quan điểm của người dùng một cách trực tiếp.
 - Có thể bộc lộ một số mô thức mà người thiết kế chưa để ý đến.
 - Đơn giản, chi phí thấp.
- Chú ý cần chọn lọc thông tin thu được.
- Hai kỹ thuật thường dùng: phỏng vấn và hỏi.

5.4.4. Kỹ thuật hỏi đáp (3) – Phỏng vấn

- Là kỹ thuật nhằm thu thập thông tin một cách trực tiếp và có cấu trúc về kinh nghiệm của người dùng với một hệ thống tương tác.
- Các câu hỏi đa dạng, phù hợp với ngữ cảnh.
- Phỏng vấn thường đi từ các câu hỏi chung đến câu hỏi chi tiết.
- Phỏng vấn cho hiệu quả khi đánh giá mức độ cao (như liệt kê các thông tin về sở thích, ấn tượng, thái độ của người dùng).
- Cần có kế hoạch trước, xác định các câu hỏi chính, các gợi ý khi phỏng vấn.
- Cần lựa chọn người phỏng vấn phù hợp với người dùng.

5.4.4. Kỹ thuật hỏi đáp (4) – Hỏi và giám sát

- Kém mềm dẻo hơn so với kỹ thuật phỏng vấn.
- Có thể sử dụng cho nhóm chủ thể rộng.
- Tốn ít thời gian.
- Một số dạng câu hỏi thường dùng:
 - Câu hỏi dạng tổng quát.
 - Câu hỏi đóng mở.
 - Câu hỏi dạng thang bậc.
 - Câu hỏi đa lựa chọn.

Chương 5 – ĐÁNH GIÁ HỆ THỐNG

5.1. Vai trò đánh giá trong thiết kế tương tác

5.2. Các kiểu đánh giá

5.3. Đánh giá thiết kế

5.4. Đánh giá cài đặt

5.5. Lựa chọn phương pháp đánh giá

5.5. Lựa chọn phương pháp đánh giá (1)

- Có nhiều kỹ thuật đánh giá, mỗi kỹ thuật đều có ưu, nhược điểm riêng.
- Các kỹ thuật này được sử dụng trong nhiều giai đoạn của vòng đời phát triển.
- Cần lựa chọn phương pháp đánh giá phù hợp.
- Lựa chọn phương pháp đánh giá trên cơ sở xem xét một số yếu tố.

5.5. Lựa chọn phương pháp đánh giá (2)

- Một số yếu tố thường được quan tâm:
 - Các giai đoạn trong vòng đời mà đánh giá thực hiện.
 - Hình thức đánh giá.
 - Mức độ chủ quan hay khách quan của kỹ thuật.
 - Kiểu số đo cung cấp.
 - Thông tin cung cấp.
 - Mức độ đan xen.
 - Tài nguyên yêu cầu.

5.5. Lựa chọn phương pháp đánh giá (3)

- Các giai đoạn trong vòng đời mà đánh giá thực hiện: đánh giá thiết kế và đánh giá cài đặt.
 - Nên đánh giá trong giai đoạn thiết kế.
 - Đánh giá thiết kế hầu như chỉ liên quan đến các chuyên gia, chủ yếu mang tính phân tích.
 - Đánh giá cài đặt xem người dùng là chủ thể và là thực nghiệm.

5.5. Lựa chọn phương pháp đánh giá (4)

- Hình thức đánh giá: đánh giá trong phòng thí nghiệm và đánh giá thực nghiệm.
 - Đánh giá trong phòng thí nghiệm: cho phép kiểm tra thực nghiệm và quan sát, không cần sự có mặt của người dùng.
 - Đánh giá thực nghiệm: tự nhiên hơn so với đánh giá trong phòng thí nghiệm.
 - Nên thực hiện cả hai kiểu đánh giá trong giai đoạn thiết kế.

5.5. Lựa chọn phương pháp đánh giá (5)

- Mức độ chủ quan và khách quan của đánh giá:
 - Một số kỹ thuật đánh giá mang tính chủ quan, như Walkthrough hay Think aloud.
 - Các kỹ thuật mang tính chủ quan sẽ dựa vào tri thức và kinh nghiệm của người đánh giá.
 - Người đánh giá phải nhận thức và hiểu được cái người dùng sẽ làm.

5.5. Lựa chọn phương pháp đánh giá (6)

- Kiểu số đo cung cấp: định lượng hay định tính.
 - Số đo định lượng: thường là số. Dễ phân tích bằng kỹ thuật thống kê.
 - Số đo định tính: thường không phải là số. Khó phân tích. Có thể cung cấp các chi tiết quan trọng không thể xác định bằng số.
 - Kỹ thuật chủ quan thường cung cấp số đo định tính. Kỹ thuật khách quan thường cung cấp số đo định lượng.

5.5. Lựa chọn phương pháp đánh giá (7)

- Thông tin cung cấp: mức độ thông tin yêu cầu cho đánh giá khá đa dạng.
- Đáp ứng tức thời:
 - Một số kỹ thuật ghi lại hành vi của người dùng tại thời điểm tương tác (như Think aloud).
 - Một số kỹ thuật liên quan đến tập sự kiện của người dùng (Walkthrough).

5.5. Lựa chọn phương pháp đánh giá (8)

- Mức độ đan xen:
 - Một số kỹ thuật có ảnh hưởng đến cách thức ứng xử của người dùng (đặc biệt là các kỹ thuật cung cấp số đo).
 - Các hành động nhạy cảm của người đánh giá có thể hỗ trợ, nhằm làm giảm hành vi nguy hiểm của người dùng.
 - Không thể cùng một lúc loại bỏ tất cả các hành vi nguy hiểm của người dùng.
 - Hầu hết các kỹ thuật đánh giá tức thời có mức độ đan xen cao (trừ nhật ký hệ thống tự động).

5.5. Lựa chọn phương pháp đánh giá (9)

- Tài nguyên yêu cầu:
 - Tài nguyên cần xem xét: thiết bị, thời gian, tiền bạc, kinh nghiệm chuyên gia, ngữ cảnh.
 - Khi tài nguyên bị hạn chế, người đánh giá phải quyết định đánh giá nào sẽ được chọn để có hiệu quả, thông tin nào có ích cho hệ thống trong điều kiện xem xét.
 - Khi kinh nghiệm của người đánh giá bị hạn chế, nên dùng các kỹ thuật heuristic, không nên sử dụng kỹ thuật phân tích.
 - Ngữ cảnh cũng có thể ảnh hưởng đến kết quả đánh giá.

Chương 6 – THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG

6.1. Tổng quan

6.2. Thiết kế giao diện người dùng hướng người dùng

6.3. Thiết kế giao diện người dùng hướng nhiệm vụ

Chương 6 – THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG

6.1. Tổng quan

6.1.1. *Giao diện người dùng*

6.1.2. *Các nguyên tắc khi thiết kế giao diện người dùng*

6.1.1. Giao diện người dùng (1)

- Giao diện người dùng là một trong những thành phần quan trọng nhất của bất kỳ một hệ thống máy tính nào.
- Lý do:
 - Giao diện người dùng cho phép người sử dụng nghe, nhìn, tiếp xúc với hệ thống.
 - Người dùng hầu như không tiếp xúc trực tiếp với các thiết kế về chức năng; chủ yếu tiếp xúc trực tiếp với các thiết bị giao tiếp (màn hình, bàn phím, chuột, ...)

6.1.1. Giao diện người dùng (2)

- Mục đích của thiết kế giao diện người dùng: cho phép người dùng có thể làm việc với máy tính một cách dễ dàng, hiệu quả và thích thú.
- Các thành phần chính của giao diện người dùng:
 - Đầu vào: bàn phím, chuột, ...
 - Đầu ra: màn hình hiển thị, âm thanh, ...

6.1.1. Giao diện người dùng (3)

- Lợi ích của việc thiết kế giao diện người dùng tốt:
 - Giảm thời gian lập trình cho sản phẩm.
 - Giảm chi phí đào tạo.
 - Giảm lỗi người dùng.
 - Tăng năng suất lao động.
 - Giúp tạo ra những sản phẩm có chất lượng cao.
 - Tăng khả năng bán được của sản phẩm.
 - Giúp giảm các bệnh nghề nghiệp.
 - Giúp giảm các lỗi nguy hiểm đến tính mạng.

6.1.2. Các nguyên tắc khi thiết kế giao diện người dùng (1)

- Một giao diện người dùng tốt cần đảm bảo các yêu cầu sau:
 - Tính thẩm mỹ.
 - Tính rõ ràng.
 - Tính tương thích.
 - Tính cấu hình.
 - Tính nhất quán.
 - Tính hiệu quả.
 - Tính trực tiếp.
 - ...

6.1.2. Các nguyên tắc khi thiết kế giao diện người dùng (2)

- Tính thẩm mỹ:
 - Tuân theo một số nguyên lý thiết kế chung:
 - + Độ tương phản cao giữa các thành phần trên màn hình.
 - + Nhóm các thành phần có cùng chức năng, nhiệm vụ.
 - + Cho phép các biểu diễn 3D.
 - Sử dụng đồ họa và màu sắc phải đơn giản, hiệu quả.
 - Phải cuốn hút được người dùng.

6.1.2. Các nguyên tắc khi thiết kế giao diện người dùng (3)

- Tính rõ ràng:
 - Giao diện phải rõ ràng về mặt hiển thị, thiết kế cũng như ngữ nghĩa:
 - + Các thành phần hiển thị.
 - + Các chức năng.
 - + Các từ và văn bản.

6.1.2. Các nguyên tắc khi thiết kế giao diện người dùng (4)

- Tính tương thích:
 - Tương thích với người sử dụng: thiết kế phải phù hợp với nhu cầu của người dùng.
 - Tương thích với nhiệm vụ: thiết kế phải đảm bảo các luồng dữ liệu và chức năng thực hiện đúng nhiệm vụ đặt ra; người dùng không phải cố gắng để thực hiện nhiệm vụ.
 - Tương thích với sản phẩm: các hệ thống mới thường được phát triển trên cơ sở các phiên bản cũ. Vì vậy, sản phẩm phải được thiết kế sao cho có thể áp dụng được các thói quen, sự hiểu biết của người dùng cũ.

6.1.2. Các nguyên tắc khi thiết kế giao diện người dùng (5)

- Tính cấu hình:
 - Thiết kế phải cho phép dễ dàng cá nhân hóa và cung cấp khả năng cấu hình được, nhằm:
 - + Nâng cao độ nhạy về mặt điều khiển.
 - + Khuyến khích người dùng chủ động tìm hiểu.

6.1.2. Các nguyên tắc khi thiết kế giao diện người dùng (6)

- Tính nhất quán:
 - Trong cùng một ngữ cảnh, một hệ thống phải được nhìn thấy, phản ứng và thao tác theo cùng một cách.
 - Cùng một thành phần phải: có cùng một diện mạo, cùng tính năng sử dụng, hoạt động tương tự nhau.
 - Với cùng một hành động, phải có cùng một kết quả.
 - Chức năng của các thành phần phải cố định.
 - Vị trí của các thành phần chuẩn hóa phải cố định.

6.1.2. Các nguyên tắc khi thiết kế giao diện người dùng (7)

- Tính hiệu quả:
 - Cần tối thiểu hóa chuyển động của tay, mắt, và các hoạt động điều khiển khác:
 - + Chuyển đổi giữa các điều khiển hệ thống phải mềm mại và dễ dàng.
 - + Các định hướng chỉ dẫn phải ngắn gọn, rõ ràng.
 - + Chuyển động của mắt phải trực diện và liên tục.
 - Nên dự báo được những mong muốn của người dùng.

6.1.2. Các nguyên tắc khi thiết kế giao diện người dùng (7)

- Tính trực tiếp:
 - Cung cấp cách thức hoàn thành nhiệm vụ một cách trực tiếp:
 - + Cần cung cấp các phương án hoàn thành nhiệm vụ.
 - + Hiện thị đáp ứng của việc tác động lên các thành phần của giao diện.

6.1.2. Các nguyên tắc khi thiết kế giao diện người dùng (8)

- Tính thân thiện:
 - Sử dụng các khái niệm và ngôn ngữ gần gũi với người dùng.
 - Giao diện nên tự nhiên, phỏng theo những mẫu hành vi của người dùng.

6.1.2. Các nguyên tắc khi thiết kế giao diện người dùng (9)

- Tính mềm dẻo:
 - Hệ thống phải nhạy với những yêu cầu khác nhau của người dùng, cho phép thực hiện các yêu cầu với những mức độ khác nhau với một hiệu quả nào đó, dựa trên:
 - + Hiểu biết và kỹ năng của người dùng.
 - + Kinh nghiệm của người dùng.
 - + Sở thích cá nhân.
 - + Thói quen.
 - + Điều kiện làm việc.

6.1.2. Các nguyên tắc khi thiết kế giao diện người dùng (10)

- Tính bỏ qua:
 - Có khả năng bỏ qua một số lỗi chung khó tránh khỏi của người dùng.
 - Có khả năng dự báo lỗi.
 - Đề phòng những lỗi trầm trọng.
 - Có các thông báo lỗi và gợi ý cách xử lý mỗi khi gặp phải lỗi.

6.1.2. Các nguyên tắc khi thiết kế giao diện người dùng (11)

- Tính khôi phục:
 - Một hệ thống phải cho phép:
 - + Hủy bỏ hoặc quay lại một số lệnh, thao tác tại thời điểm trước đó.
 - + Quay lại một số bước trước đó nếu có khó khăn xảy ra.
 - Đảm bảo người dùng không bao giờ bị mất kết quả tại trạng thái hiện tại.

6.1.2. Các nguyên tắc khi thiết kế giao diện người dùng (12)

- Tính trách nhiệm:
 - Hệ thống phải trả lời nhanh những yêu cầu của người dùng.
 - Hệ thống phải cung cấp phản hồi tức thì đối với mọi hành động của người dùng, thông qua:
 - + Hình ảnh.
 - + Văn bản.
 - + Âm thanh.

6.1.2. Các nguyên tắc khi thiết kế giao diện người dùng (13)

- Tính đơn giản:
 - Cung cấp giao diện đơn giản nhất có thể.
 - Một số cách thức tạo ra sự đơn giản:
 - + Sử dụng các menu dạng ẩn, chỉ xuất hiện khi cần thiết.
 - + Hiện thị các chức năng chung và cần thiết trước tiên.
 - + Cung cấp các mặc định.
 - + Tạo các hành động chung, đơn giản.

6.1.2. Các nguyên tắc khi thiết kế giao diện người dùng (14)

- Tính trong suốt:
 - Cho phép người dùng tập trung vào nhiệm vụ, tránh tập trung vào các cơ chế hoạt động của giao diện.
 - Không để người dùng tập trung vào các chi tiết kỹ thuật của hệ thống.
 - Các suy nghĩ của người dùng phải hướng đến nhiệm vụ.

6.1.2. Các nguyên tắc khi thiết kế giao diện người dùng (15)

- Tính dễ hiểu, dễ học:
 - Một hệ thống phải dễ hiểu, dễ học. Người dùng phải biết:
 - + Nhìn vào đâu.
 - + Phải làm gì, khi nào thì làm, làm ở đâu.
 - + Tại sao phải thực hiện.
 - + Cách thức thực hiện.
 - Chuỗi hành động, đáp ứng và trình diễn thông tin phải theo trật tự có khả năng dự đoán được.

Chương 6 – THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG

6.1. Tổng quan

6.2. Thiết kế giao diện người dùng hướng người dùng

6.3. Thiết kế giao diện người dùng hướng nhiệm vụ

Chương 6 – THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG

6.2. Thiết kế giao diện người dùng hướng người dùng

6.2.1. *Khái niệm*

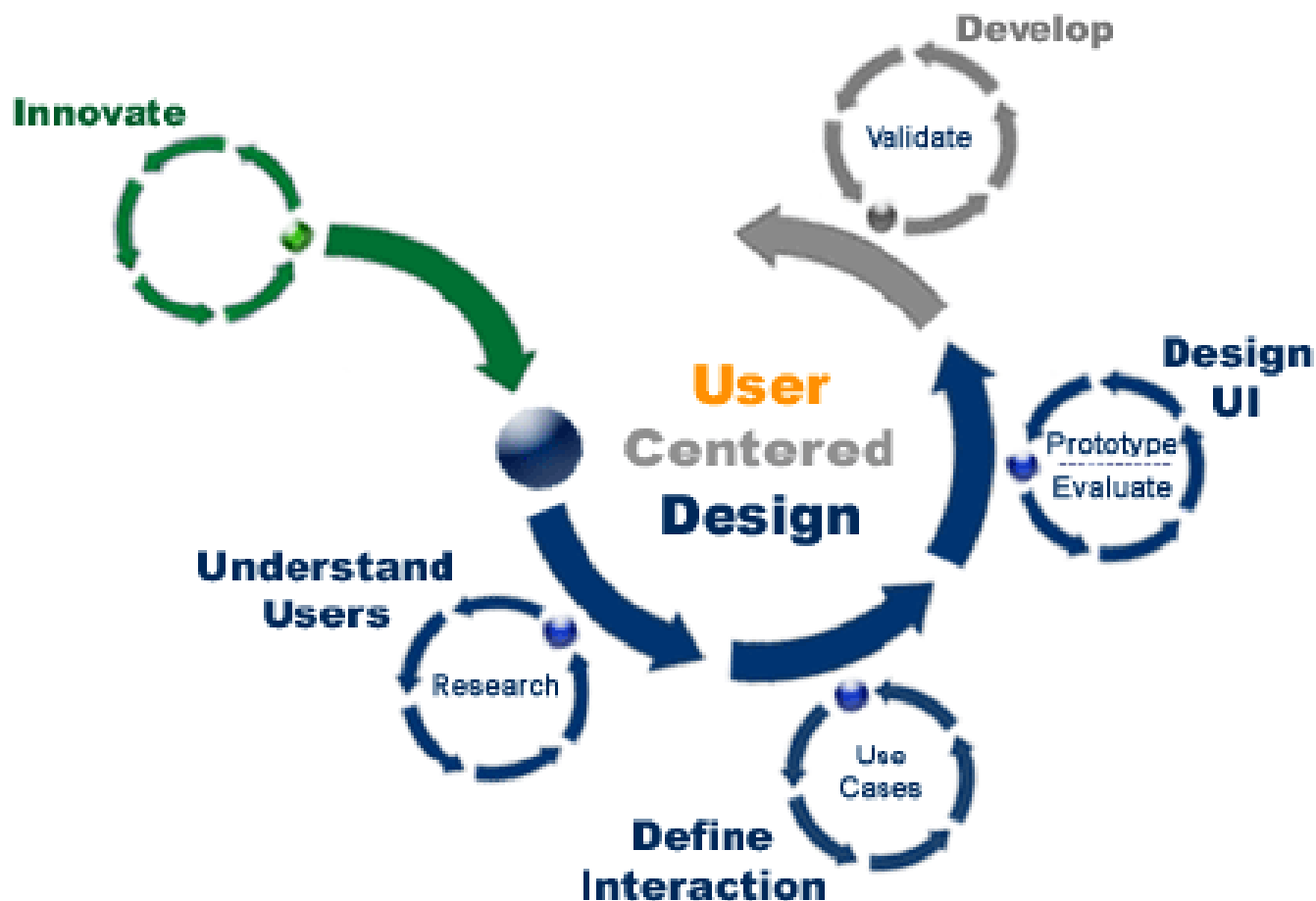
6.2.2. *Quy trình thiết kế*

6.2.3. *Một số phương pháp thiết kế hướng người dùng*

6.2.1. *Khái niệm (1)*

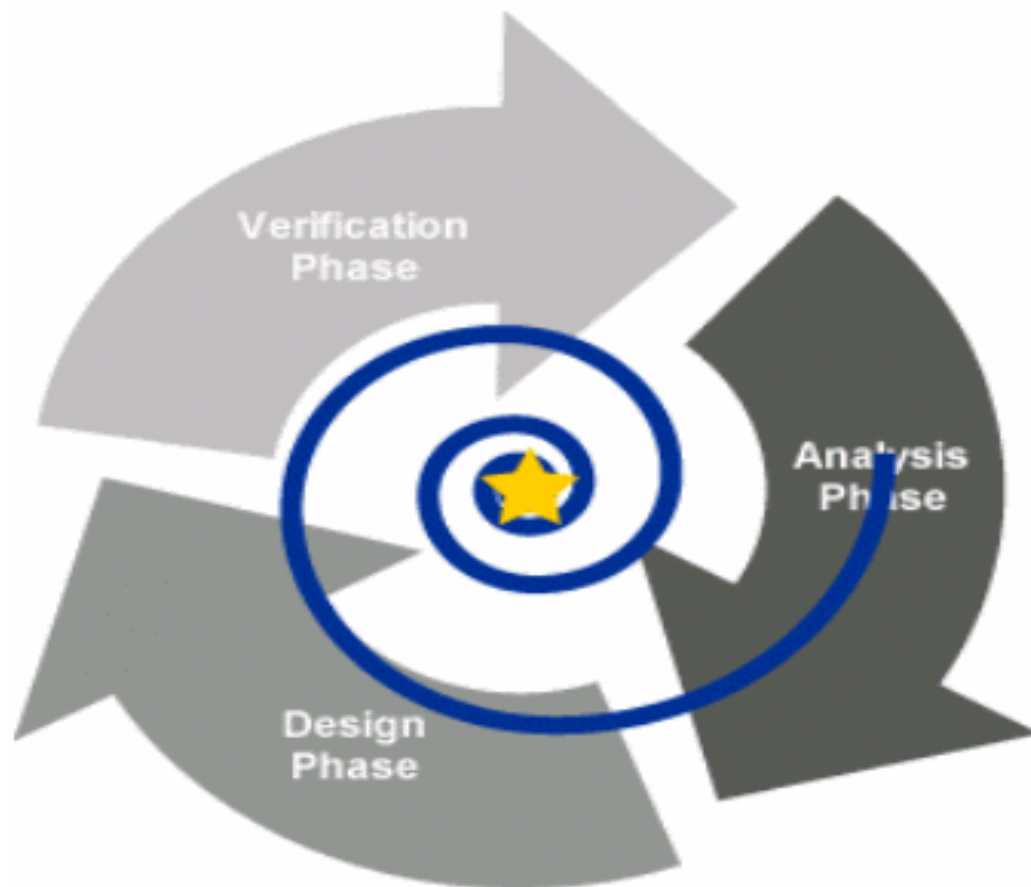
- Thiết kế hướng người dùng: lấy người dùng làm trung tâm.
 - Đặt yêu cầu của người dùng vào trung tâm của quá trình thiết kế và phát triển.
 - Tiến hành thử nghiệm và đánh giá với người dùng.
 - Quy trình thiết kế là thiết kế lặp.

6.2.1. Khái niệm (2)



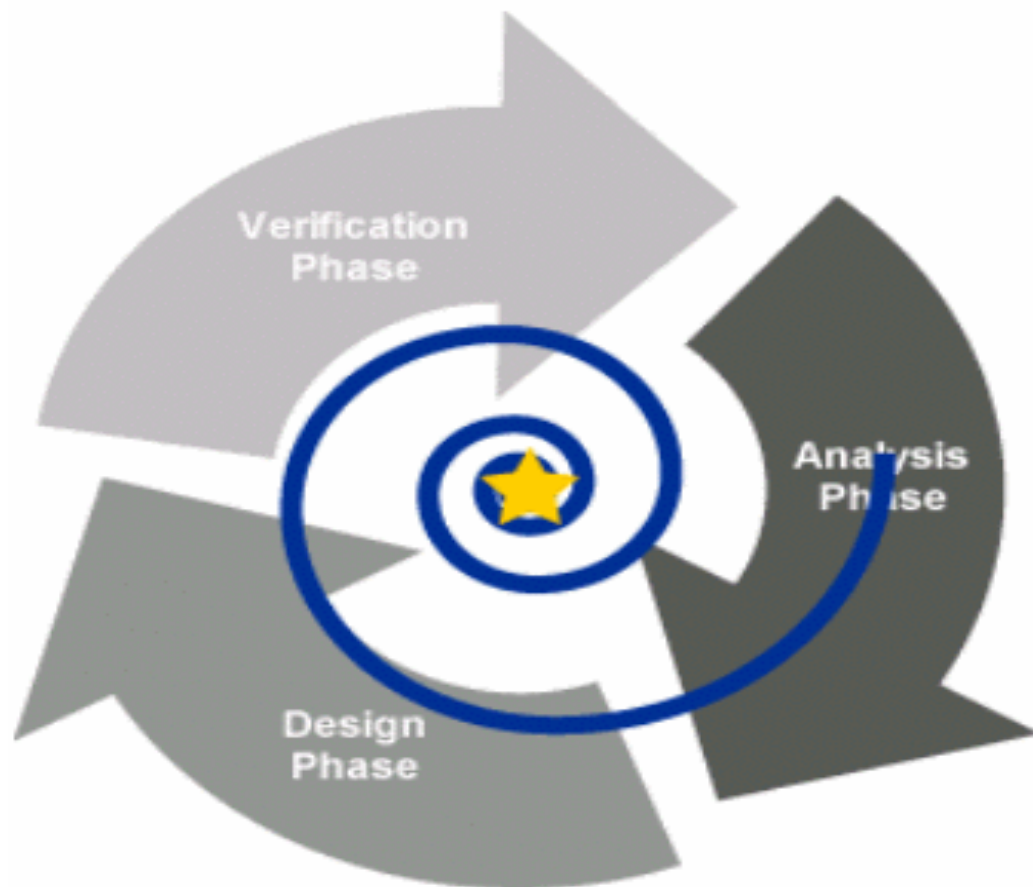
6.2.2. Quy trình thiết kế hướng người dùng (1)

- Gồm 3 pha:
 - Pha phân tích
 - Pha thiết kế
 - Pha đánh giá



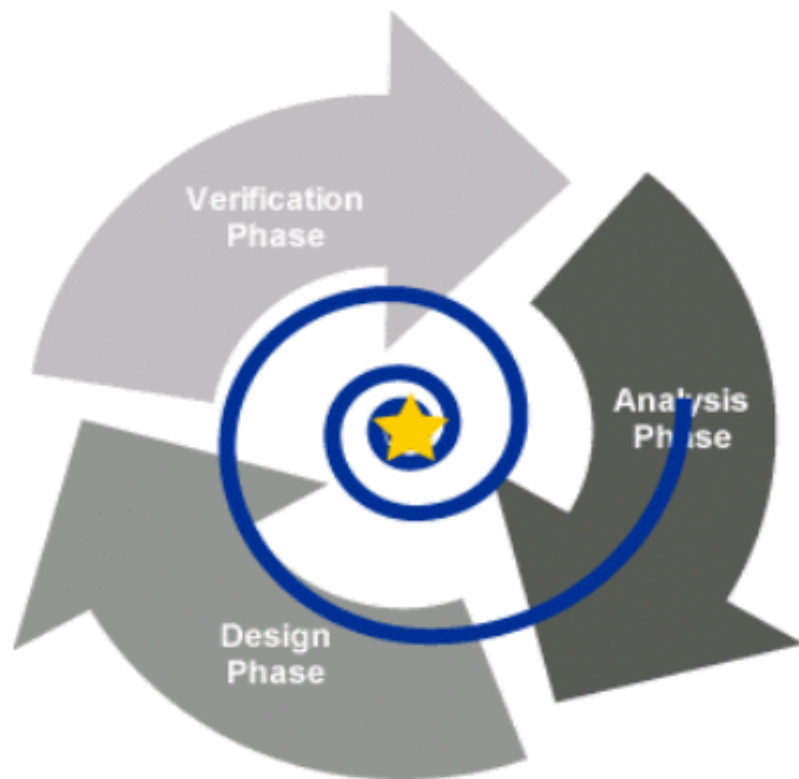
6.2.2. Quy trình thiết kế hướng người dùng (2)

- Quy trình thiết kế là thiết kế lặp:
 - Việc đánh giá, kiểm thử các giả thuyết, các khái niệm, các thiết kế được diễn ra liên tục với người sử dụng.
 - Mỗi pha được thực hiện không chỉ một lần (từ 2 đến 3 lần).



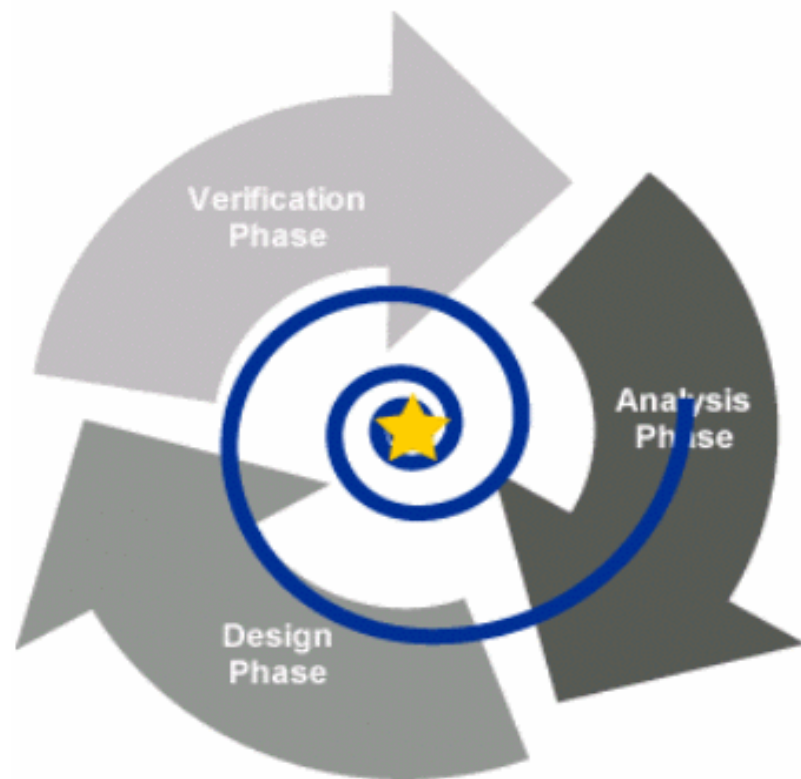
6.2.2. Quy trình thiết kế hướng người dùng (3)

- Phân tích:
 - Đặc tả yêu cầu người dùng.
 - Nghiên cứu thị trường.
 - Tìm hiểu các đối thủ cạnh tranh.
 - Tìm hiểu về người sử dụng.
 - Nghiên cứu, lựa chọn công nghệ.



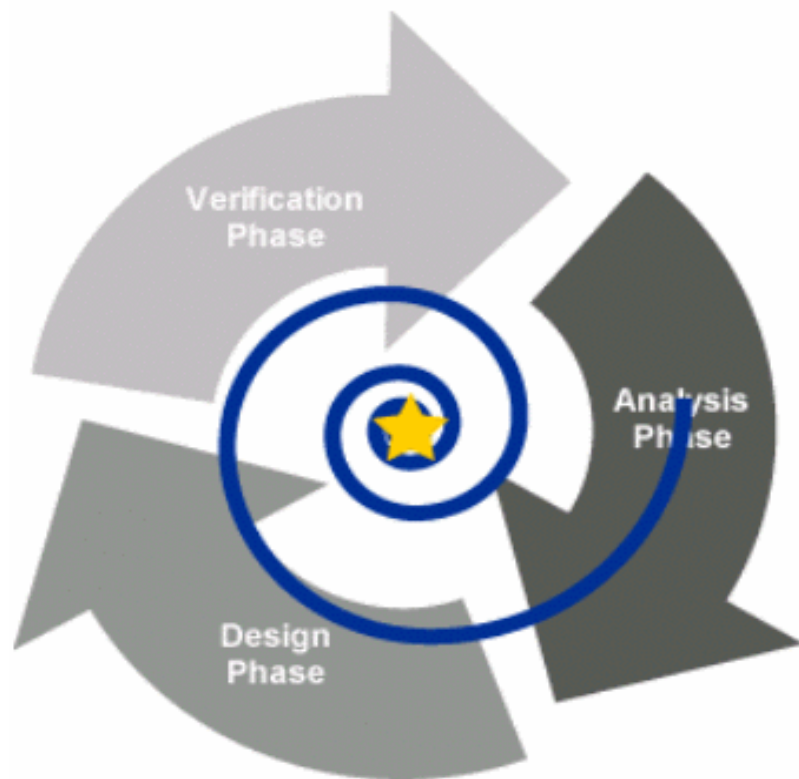
6.2.2. Quy trình thiết kế hướng người dùng (4)

- Thiết kế:
 - Tổ chức các chức năng.
 - Xây dựng mô hình định hướng về mặt kiến trúc.
 - Phát triển các khái niệm hàm.
 - Xây dựng bản mẫu.



6.2.2. Quy trình thiết kế hướng người dùng (5)

- Đánh giá:
 - Đánh giá giá trị chức năng.
 - Đánh giá giá trị hình thức.
 - Áp dụng các tiêu chuẩn để đánh giá thiết kế tổng thể.
 - Đánh giá thiết kế tổng thể so với các đối thủ cạnh tranh.



6.2.2.1. Pha phân tích (1)

- Đặc tả yêu cầu người dùng:
 - Để xác định mục đích và đối tượng của thiết kế.
 - Cần trả lời các câu hỏi sau:
 - + Xác định người dùng là ai.
 - + Mục đích của họ là gì.
 - + Nhiệm vụ nào họ muốn hoàn thành.

6.2.2.1. *Pha phân tích (2)*

- Nghiên cứu thị trường: để có được sự hiểu biết về thị trường, cần nắm được các thông tin sau:
 - Loại đối tượng khách hàng.
 - Phạm vi thị trường.
 - Thị trường tiêu thụ mục tiêu.

6.2.2.1. Pha phân tích (3)

- Tìm hiểu các đối thủ cạnh tranh: để hiểu về các đối thủ cạnh tranh, cần tìm hiểu các thông tin sau:
 - Đối thủ cạnh tranh là ai.
 - Sản phẩm mà họ cung cấp cho người sử dụng có giá trị thế nào về mặt chức năng.
 - Đánh giá của người sử dụng về sản phẩm của họ như thế nào (thích hay không thích ở những điểm nào).
 - Các chức năng còn thiếu sót trong sản phẩm của họ.
 - Thị trường mà mình chắc chắn có ưu thế cạnh tranh hơn.

6.2.2.1. Pha phân tích (4)

- Tìm hiểu về người sử dụng:
 - Vai trò của họ trong tổ chức.
 - Nền tảng của người sử dụng (nền tảng giáo dục, kinh nghiệm, ...)
 - Các khả năng tự nhiên của người sử dụng (khả năng nghe nhìn, ...)

6.2.2.1. Pha phân tích (5)

- Nghiên cứu, lựa chọn công nghệ:
 - Phải nắm được các công nghệ sẵn có hiện tại.
 - Lựa chọn công nghệ phù hợp với mục đích.
 - Cần theo kịp xu hướng phát triển của công nghệ.

6.2.2.1. Pha phân tích (6)

- Các kỹ thuật phân tích thường được sử dụng:
 - Phân tích người sử dụng:
 - + Sử dụng các mô hình người dùng
 - + Mô tả người dùng

6.2.2.1. Pha phân tích (7)

- Các kỹ thuật phân tích thường được sử dụng:
 - Phân tích nhiệm vụ:
 - + Xây dựng các use – case cần thiết
 - + Phân tích nhiệm vụ phân cấp
 - + Phân tích nhiệm vụ theo nhận thức (GOMS, TAG, ...)

6.2.2.1. *Pha phân tích (8)*

- Các kỹ thuật phân tích thường được sử dụng:
 - Phân tích thị trường:
 - + Phân tích nhu cầu
 - + Phân tích cạnh tranh
 - + Phân tích ảnh hưởng tài chính

6.2.2.2. *Pha thiết kế (1)*

- Các bước trong pha thiết kế là khác nhau, phụ thuộc vào số lần lặp trong chu trình.

6.2.2.2. Pha thiết kế (2)

- Các bước lập đầu tiên:
 - Tổ chức các chức năng đã tập hợp trong pha phân tích.
 - Xây dựng một mô hình kiến trúc có thể hoạt động được và có thể kiểm soát được, bằng cách:
 - + Xây dựng bản mẫu phác thảo để mô phỏng sơ đồ chức năng mức cao, cấu trúc nhiệm vụ mức cao.
 - + Dựa trên bản mẫu đó, đánh giá các thay đổi, các giả thuyết thiết kế để đảm bảo nền tảng giao diện thiết kế đáp ứng được nhu cầu của người sử dụng.

6.2.2.2. Pha thiết kế (3)

- Các bước lặp sau:
 - Tập trung chỉnh sửa các chi tiết (cách thức trình diễn, thứ tự và các chi tiết xuất hiện trong mỗi màn hình giao diện đang thiết kế.
 - Xây dựng các bản mẫu có tính chân thực hơn.
 - Kiểm thử đối với một số tương tác cụ thể để đảm bảo sản phẩm cuối cùng có thể đáp ứng các chức năng, nhiệm vụ đã được đặc tả.

6.2.2.3. Pha đánh giá (1)

- Chuẩn bị và thực hiện các kiểm thử về tính tiện dụng của sản phẩm.
- Báo cáo kết quả kiểm thử cho nhóm thiết kế.
- Các kiểm thử này được sử dụng để đánh giá các giả thiết và các khái niệm thiết kế; tập hợp các thông tin cần thiết.
- Loại và độ phức tạp của các trường hợp kiểm thử là khác nhau, phụ thuộc vào số lần lặp lại trong chu trình phát triển.

6.2.2.3. Pha đánh giá (2)

- Các lần lặp đầu tiên:
 - Kiểm thử nhằm mục đích:
 - + Thu thập thông tin (về thị trường, các đối thủ cạnh tranh, chức năng của hệ thống, người sử dụng, các nhiệm vụ).
 - + Kiểm định lại các giả thiết thiết kế.

6.2.2.3. Pha đánh giá (3)

- Các lần lặp đầu tiên:
 - Sử dụng các phương pháp đơn giản:
 - + Gửi danh sách câu hỏi.
 - + Phỏng vấn trực tiếp.
 - + Tổ chức nhóm thảo luận.
 - + Giám sát quá trình hoạt động thực tế.

6.2.2.3. Pha đánh giá (4)

- Các lần lặp tiếp theo:
 - Kiểm thử nhằm mục đích kiểm định lại các quyết định thiết kế.
 - Quan sát người dùng thực hiện các nhiệm vụ khác nhau để xác định họ có thực sự biết cách sử dụng các giao diện được thiết kế hay không.
 - Có thể sử dụng các phương pháp kiểm thử phức tạp:
 - + Cùng học để khám phá.
 - + Kiểm thử theo chuẩn.

6.2.3. Một số phương pháp thiết kế hướng người dùng

- Nhóm tập trung.
- Kiểm thử tính dùng được.
- Sắp xếp phiếu đánh giá.
- Thiết kế hợp tác.
- Lập bảng câu hỏi.
- Phỏng vấn.
- Nên kết hợp các phương pháp này để thực hiện tương tác với người dùng trong suốt quá trình tìm hiểu, phân tích, thiết kế, cài đặt thử nghiệm, vận hành và khai thác.

6.2.3.1. Nhóm tập trung

- Một nhóm người dùng được mời đến để cùng chia sẻ suy nghĩ, cảm nhận, thái độ, ý kiến về một chủ đề nào đó.
- Kết quả thảo luận được sử dụng làm đầu vào thiết kế.
- Kết quả này thường là các dữ liệu không có tính thống kê.
- Để việc thảo luận có hiệu quả, cần có một nhóm trưởng có kinh nghiệm và một nhà phân tích.

6.2.3.2. Kiểm thử tính dùng được (1)

- Thu thập dữ liệu từ những người đã sử dụng sản phẩm để đánh giá một vấn đề nào đó của sản phẩm.
- Mời một người tham gia một phiên kiểm thử.
- Một người điều khiển yêu cầu người dùng thực hiện một số nhiệm vụ.
- Một người ghi chép lại tất cả những khó khăn mà người dùng gặp phải.
- Người dùng được hỏi họ đang làm gì và vì sao.
- Đo thời gian người dùng hoàn thành một nhiệm vụ.

6.2.3.2. Kiểm thử tính dùng được (2)

- Phương pháp này thường được sử dụng:
 - Làm đầu vào của quá trình thiết kế.
 - Ở giai đoạn kiểm tra mẫu thử.
 - Khi kết thúc dự án.
- Đây là cách tốt nhất để tìm ra những khó khăn khi sử dụng sản phẩm.
- Kết quả có thể là những thông tin mang tính thống kê hoặc không.

6.2.3.3. Sắp xếp phiếu đánh giá (1)

- Sử dụng nhóm chuyên gia hoặc người dùng thử không có kinh nghiệm trong thiết kế.
- Hoạt động có thể tổ chức theo nhóm hoặc tiến hành với từng cá nhân.
- Xác định các khái niệm đầu tiên và viết các khái niệm này trên các thẻ có đánh số.
- Những người tham gia được hướng dẫn và yêu cầu tổ chức các thẻ này thành một cấu trúc cây.

6.2.3.3. Sắp xếp phiếu đánh giá (2)

- Phương pháp này thường được sử dụng làm đầu vào cho thiết kế.
- Đây là một cách thức tốt để phân loại nội dung (các công việc, cấu trúc menu, ...) và đưa ra kiến trúc về mặt thông tin.
- Có thể sử dụng phương pháp này để tạo ra các dữ liệu thống kê.

6.2.3.4. Thiết kế hợp tác

- Không chỉ đòi hỏi ý kiến của người dùng về thiết kế, mà còn yêu cầu họ tham gia vào quá trình thiết kế cũng như quá trình ra quyết định.
- Để hiệu quả, cần có một người điều khiển có kinh nghiệm để điều tiết các tri thức và hướng dẫn người dùng cùng tham gia.
- Thường sử dụng đối với các dự án nhỏ, nhằm đưa ra các mẫu thử cho quá trình thiết kế tổng thể.

6.2.3.5. *Lập bảng câu hỏi*

- Hỏi người dùng về một vấn đề nào đó, dựa trên một tập các câu hỏi đã được định nghĩa sẵn.
- Việc đặt câu hỏi phải đặc trưng, khách quan.
- Bộ câu hỏi phải cho phép phân tích thống kê các kết quả thu được.
- Có thể cho phép người sử dụng ở xa tham gia vào quá trình này.
- Số lượng mẫu người dùng thử có thể lớn.
- Thường được quản lý bằng các phương tiện điện tử.
- Là một cách tốt để tạo thông tin thống kê.

6.2.3.6. Phỏng vấn (1)

- Một người tiến hành phỏng vấn một người khác.
- Quan điểm cá nhân của người dùng thử sẽ được khai thác và ghi nhận.
- Nhanh chóng sửa chữa được những hiểu lầm giữa người phỏng vấn và người được phỏng vấn.
- Đòi hỏi người phỏng vấn và phân tích có kinh nghiệm.

6.2.3.6. Phỏng vấn (2)

- Kết quả của quá trình này có thể là những thông tin phi thống kê.
- Các thông tin này sẽ được nghiên cứu phân tích bởi các chuyên viên có kinh nghiệm.
- Phương pháp này thường được thực hiện trước quá trình thiết kế nhằm thu thập các thông tin, những tri thức về lĩnh vực hoạt động hay những yêu cầu cụ thể.

Chương 6 – THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG

6.1. Tổng quan

6.2. Thiết kế giao diện người dùng hướng người dùng

6.3. Thiết kế giao diện người dùng hướng nhiệm vụ

Chương 6 – THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG

6.3. Thiết kế giao diện người dùng hướng nhiệm vụ

6.3.1. Quy trình thiết kế

6.3.2. Phân tích nhiệm vụ

6.3.3. Mô hình hóa nhiệm vụ

6.3.4. Đặc tả công nghệ dùng trong thiết kế

6.3.5. Đánh giá và kiểm thử

6.3.1. Quy trình thiết kế (1)

- Quy trình thiết kế hướng nhiệm vụ được xây dựng dựa theo các nhiệm vụ cụ thể mà người dùng muốn thực hiện với hệ thống.
- Những nhiệm vụ được lựa chọn ban đầu trong quá trình thiết kế sau đó được sử dụng để:
 - Xây dựng các vấn đề khác.
 - Trợ giúp cho các quyết định thiết kế.
 - Đánh giá thiết kế khi thiết kế được phát triển.

6.3.1. Quy trình thiết kế (2)

- Quy trình: gồm 4 bước chính:
 - Phân tích tình huống nhiệm vụ hiện tại.
 - Xây dựng các nhiệm vụ tương lai mà hệ thống cần thực hiện.
 - Đặc tả công nghệ dùng trong thiết kế.
 - Đánh giá và kiểm thử.

6.3.1.1. Xác định mục đích sử dụng hệ thống của người dùng (1)

- Còn gọi là “Phân tích người dùng và nhiệm vụ”.
- Phân tích người dùng và nhiệm vụ là một yêu cầu cần thiết, để đảm bảo:
 - Hệ thống xây dựng nên đúng là hệ thống cần thiết kế.
 - Hệ thống xây dựng nên hoạt động tốt và đáp ứng được công việc của người sử dụng.
- Để phân tích người dùng và nhiệm vụ hiệu quả, đòi hỏi sự tiếp xúc gần gũi giữa cá nhân các thành viên trong nhóm thiết kế và người sử dụng trực tiếp hệ thống.

6.3.1.1. Xác định mục đích sử dụng hệ thống của người dùng (2)

- Hiểu về người sử dụng cũng là một yêu cầu quan trọng.
- Cần phân tích người dùng về những vấn đề: nền tảng tri thức, sở thích, thói quen ...
- Các thông tin đó sẽ hỗ trợ người thiết kế giải quyết các vấn đề như:
 - Đặt tên cho các hệ thống menu.
 - Các chức năng, trợ giúp cần có.
 - Các đặc điểm mà hệ thống cung cấp.

6.3.1.2. Lựa chọn nhiệm vụ

- Người thiết kế nhận dạng một số nhiệm vụ cần thực hiện.
- Đó là những nhiệm vụ mà người sử dụng đã mô tả với người thiết kế.
- Những nhiệm vụ được nhận dạng ban đầu đó là những nhiệm vụ thực sự mà người dùng phải đối mặt.
- Nhóm thiết kế cần thu thập các tài liệu để thực hiện các nhiệm vụ đó.
- Những nhiệm vụ được lựa chọn nên bao trùm toàn bộ chức năng của hệ thống.

6.3.1.3. Sao chép

- Nên tham khảo những thiết kế sẵn có.
- Xây dựng ý tưởng giao diện về hệ thống của mình từ những giao diện sẵn có.
- Những đặc điểm được sao chép thường là những đặc điểm quen thuộc với người sử dụng, do đó sẽ khiến người dùng dễ dàng sử dụng hệ thống.

6.3.2. Phân tích nhiệm vụ

- Phân tích nhiệm vụ là quá trình thu thập dữ liệu về những nhiệm vụ mà người sử dụng thực hiện và phân tích để có được những hiểu biết sâu sắc về chúng.
- Các phương pháp thường dùng:
 - Phân tích nhiệm vụ phân cấp HTA
 - Phân tích nhiệm vụ dựa trên mô hình nhận thức GOMS

6.3.3. Mô hình hóa nhiệm vụ (1)

- Là hoạt động biến đổi các nhiệm vụ thô và các dữ liệu liên quan đến người sử dụng hoặc các ý tưởng thành các đơn vị có cấu trúc của tri thức nhiệm vụ.
- Để ghi lại tri thức này, sử dụng các phương pháp biểu diễn khác nhau.
- Phương pháp biểu diễn phổ biến nhất: UML (Unified Modeling Language – Ngôn ngữ mô hình hóa thống nhất)

6.3.3. Mô hình hóa nhiệm vụ (2)

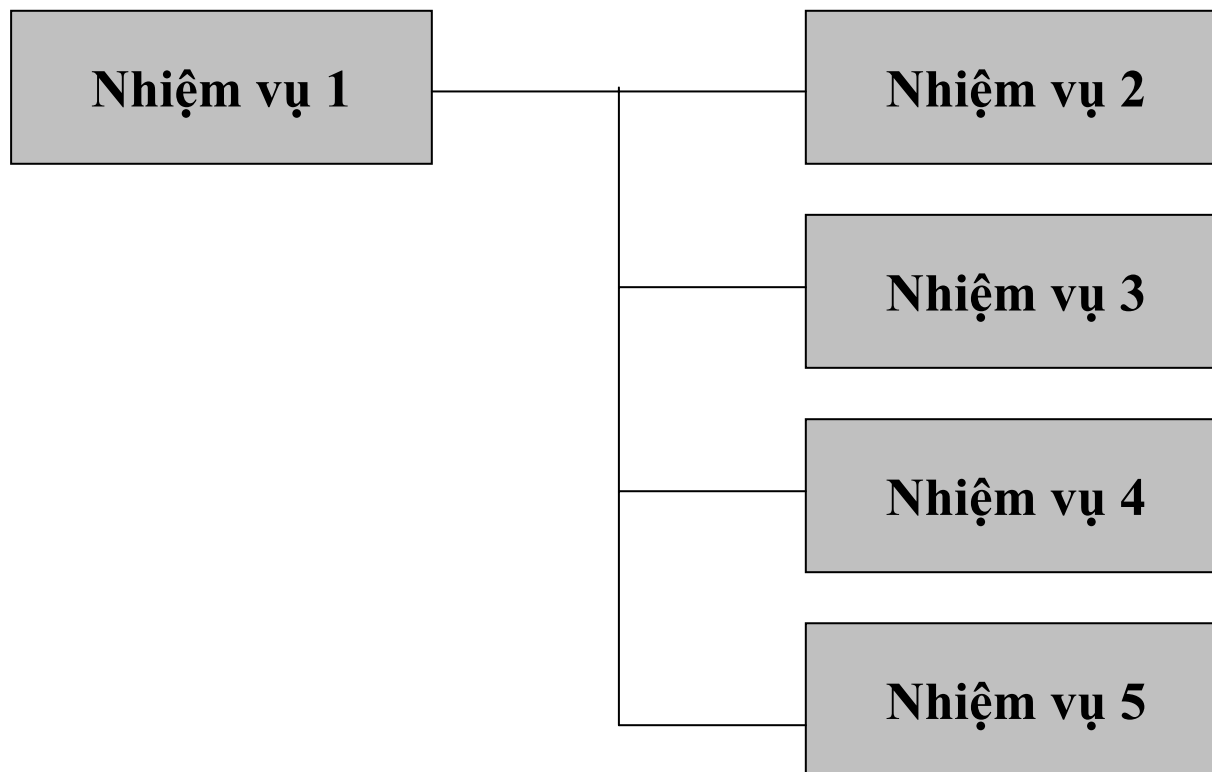
- Mục đích của việc biểu diễn các tri thức nhiệm vụ:
 - Viết tài liệu và giúp trao đổi kiến thức giữa các nhà thiết kế.
 - Phân tích công việc, tìm ra những công việc còn tồn đọng, tìm ra các cơ hội.
 - Tổ chức lại ý tưởng của các các nhân trong nhóm thiết kế.
 - Thảo luận các khía cạnh của nhiệm vụ trong nhóm thiết kế.
 - Đề nghị các thay đổi hoặc bổ xung trong nhóm thiết kế.
 - Lựa chọn các giải pháp thay thế trong nhóm thiết kế hoặc với khách hàng.

6.3.3.1. Mô hình nhiệm vụ truyền thống (1)

- Chủ yếu sử dụng cấu trúc cây nhiệm vụ để biểu diễn.
- Cây nhiệm vụ chỉ rõ cấu trúc công việc dưới dạng: nhiệm vụ, mục đích và hành động.
- Chủ yếu dựa trên các mối quan hệ nhiệm vụ con giữa các nhiệm vụ.
- Dễ hiểu, dễ xây dựng, có thể không cần phần mềm hỗ trợ.
- Hai cách mô tả: mô tả cây từ trái sang phải; mô tả cây từ trên xuống dưới.

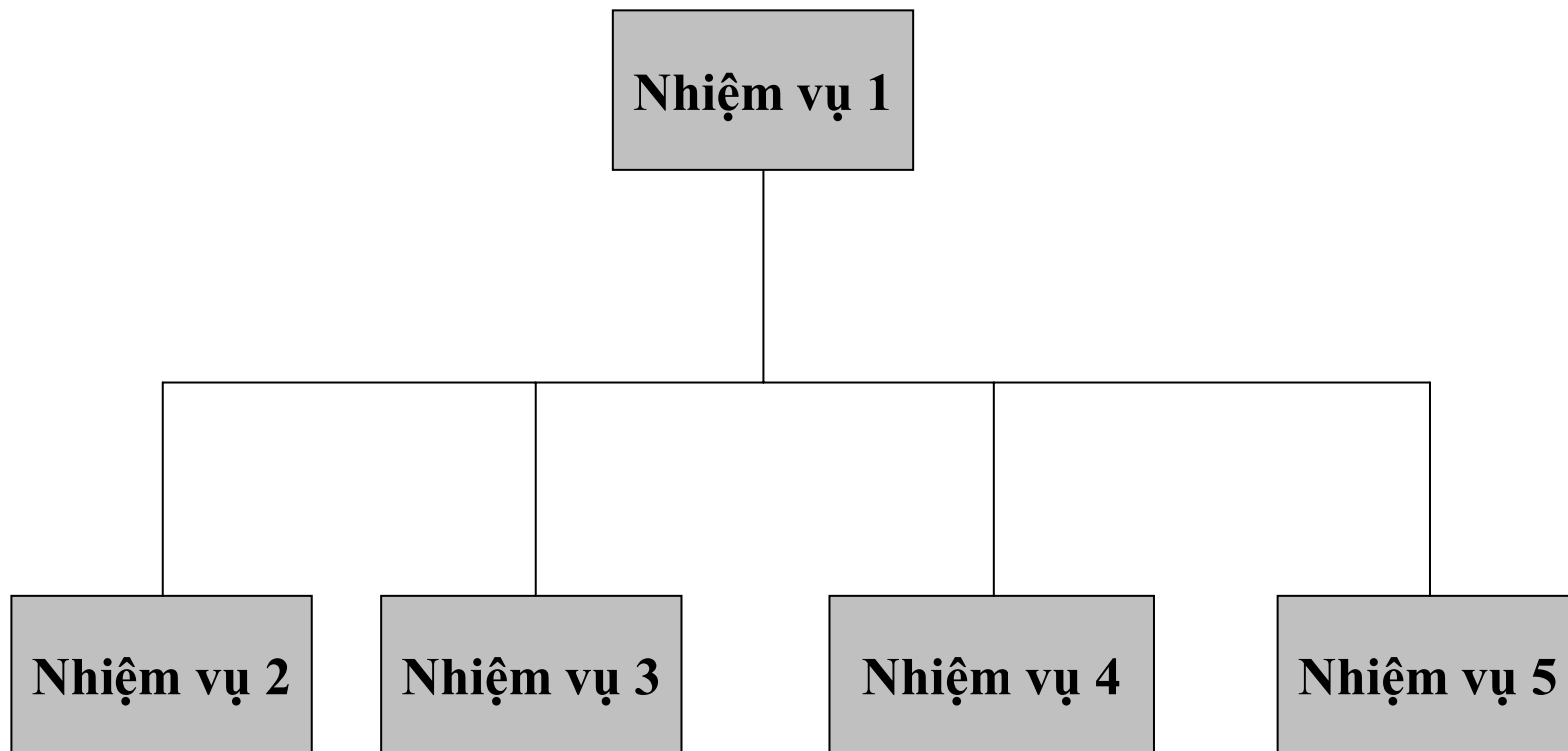
6.3.3.1. Mô hình nhiệm vụ truyền thống (2)

- Mô tả cây từ trái sang phải:



6.3.3.1. Mô hình nhiệm vụ truyền thống (3)

- Mô tả cây từ trên xuống dưới:



6.3.3.2. Ngôn ngữ mô hình hóa thống nhất UML (1)

- UML là ngôn ngữ mô hình hóa thống nhất, bao gồm các ký hiệu hình học, được sử dụng để thể hiện và miêu tả các thiết kế của một hệ thống.
- UML có thể được sử dụng làm công cụ giao tiếp giữa người dùng, nhà phân tích, nhà thiết kế và nhà phát triển sản phẩm.
- UML có thể sử dụng trong nhiều giai đoạn, từ phát triển, thiết kế cho tới thực hiện, bảo trì.
- UML không được thiết kế nhằm mục đích mô hình hóa nhiệm vụ, nhưng một số biểu đồ của UML rất hữu dụng trong việc phân tích nhiệm vụ.

6.3.3.2. Ngôn ngữ mô hình hóa thống nhất UML (2)

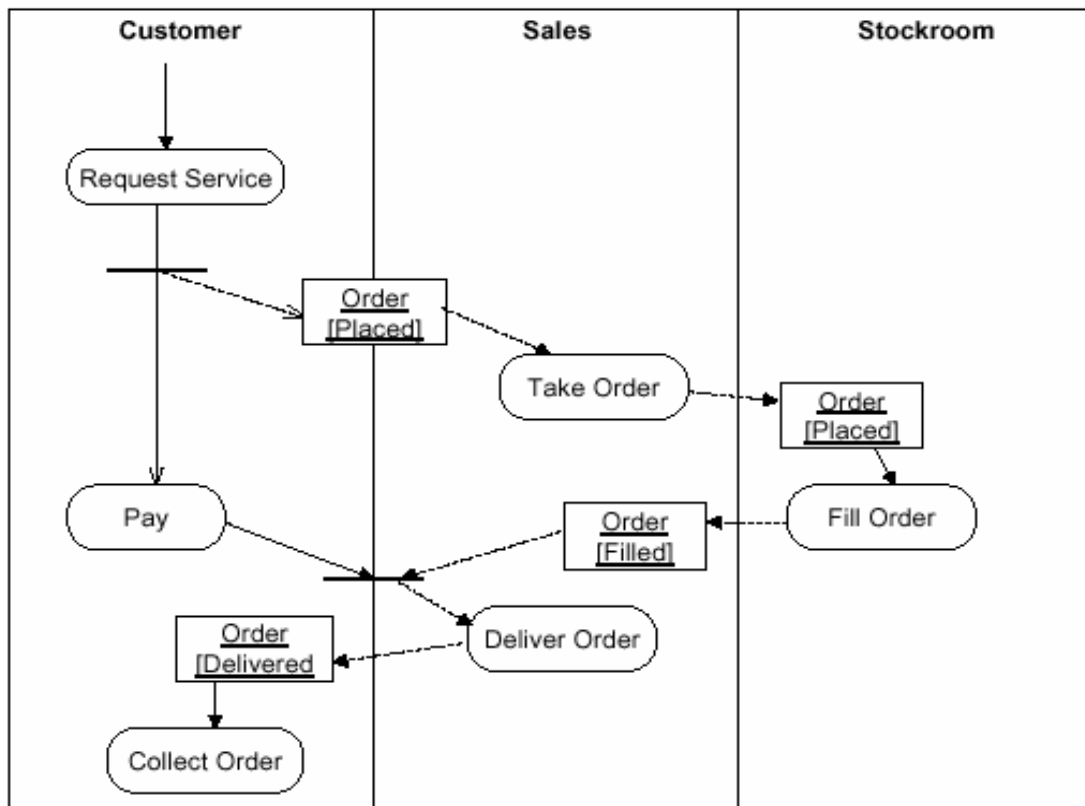
- Một số loại biểu đồ của UML:
 - Biểu đồ hoạt động.
 - Biểu đồ hợp tác.
 - Biểu đồ tuần tự.
 - Biểu đồ Use case.

6.3.3.2. Ngôn ngữ mô hình hóa thống nhất UML (3)

- Biểu đồ hoạt động:
 - Được sử dụng để mô tả luồng nhiệm vụ trong mối quan hệ với các sự kiện, vai trò, và mục đích.
 - Một hoạt động được khởi tạo bằng một sự kiện.
 - Ngay sau khi một nhiệm vụ được bắt đầu, kết quả là một mục đích được thực hiện.

6.3.3.2. Ngôn ngữ mô hình hóa thống nhất UML (4)

- Biểu đồ hoạt động cho hệ thống đặt phòng:

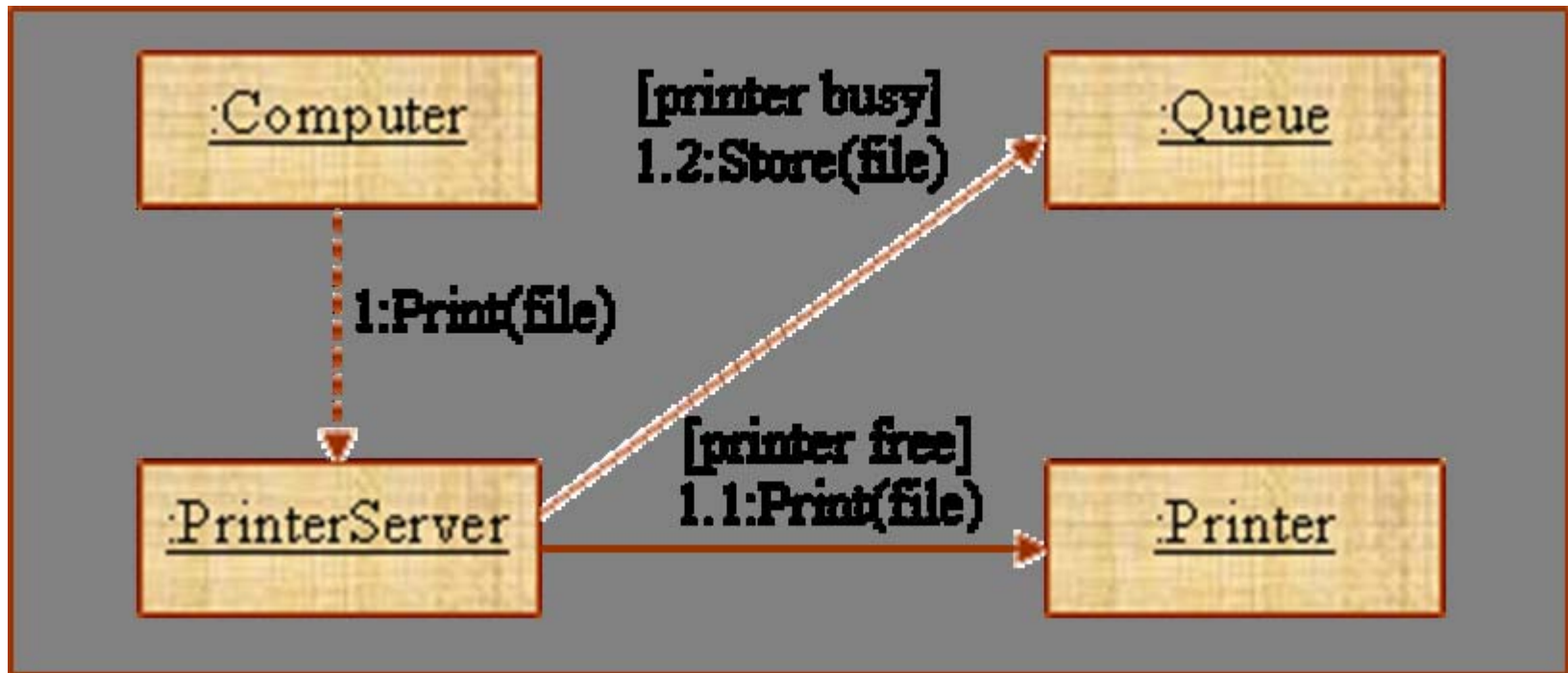


6.3.3.2. Ngôn ngữ mô hình hóa thống nhất UML (5)

- Biểu đồ hợp tác:
 - Miêu tả cách thức hợp tác của các đối tượng khác nhau.
 - Ký hiệu mũi tên chỉ sự liên lạc hay hợp tác giữa các vai trò trong khi trao đổi các đối tượng hay các thông điệp.

6.3.3.2. Ngôn ngữ mô hình hóa thống nhất UML (6)

- Biểu đồ hợp tác của Printer Server:

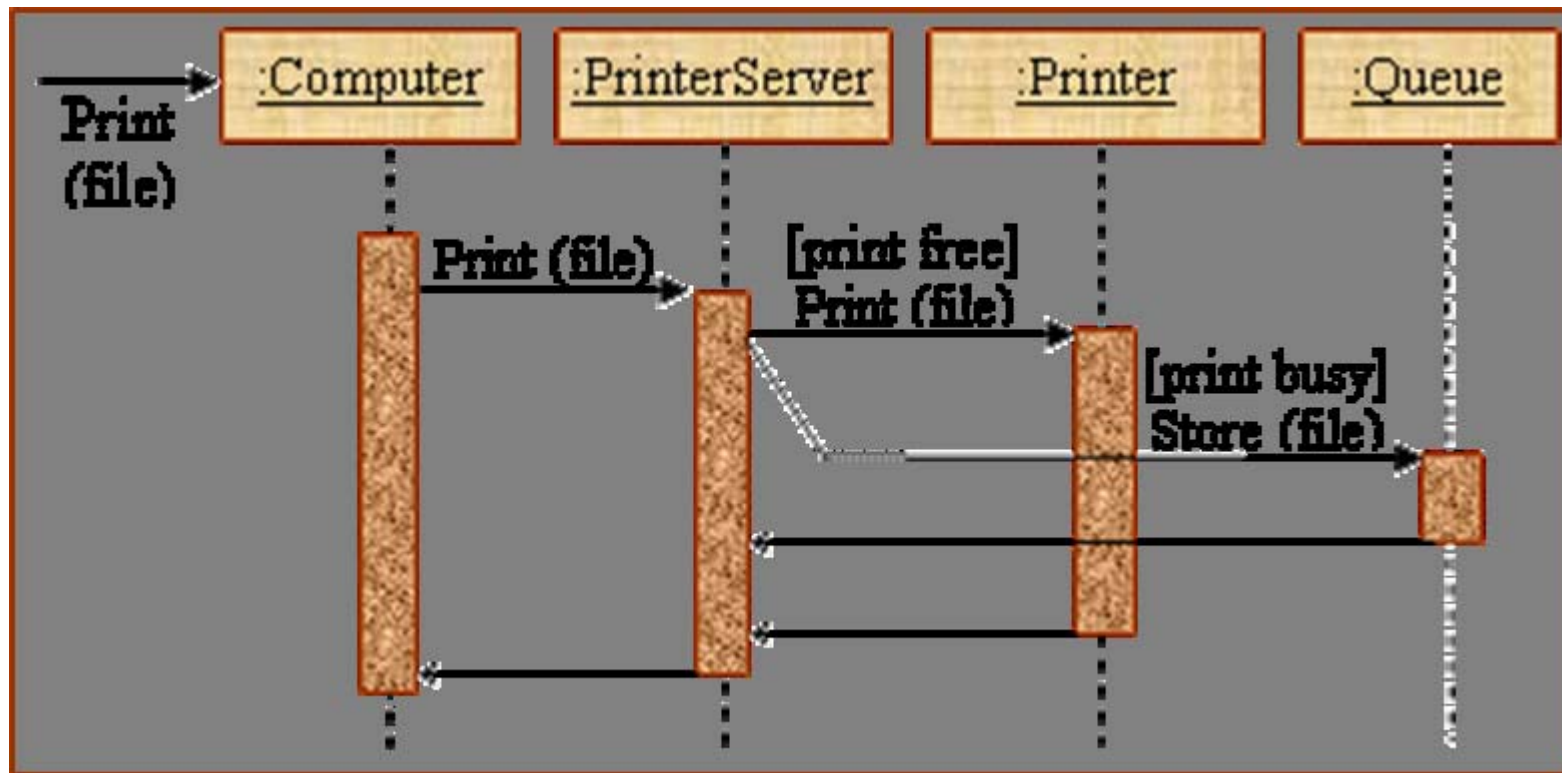


6.3.3.2. Ngôn ngữ mô hình hóa thống nhất UML (7)

- Biểu đồ tuần tự:
 - Biểu diễn thứ tự thực hiện các nhiệm vụ.
 - Biểu đồ trình tự thường chứa một loạt các đối tượng được biểu diễn bằng các đường thẳng đứng.
 - Các thông điệp được biểu diễn bằng các đường gạch ngang gắn liền với các mũi tên nối giữa các đường thẳng đứng.

6.3.3.2. Ngôn ngữ mô hình hóa thống nhất UML (8)

- Biểu đồ tuần tự cho Printer Server:

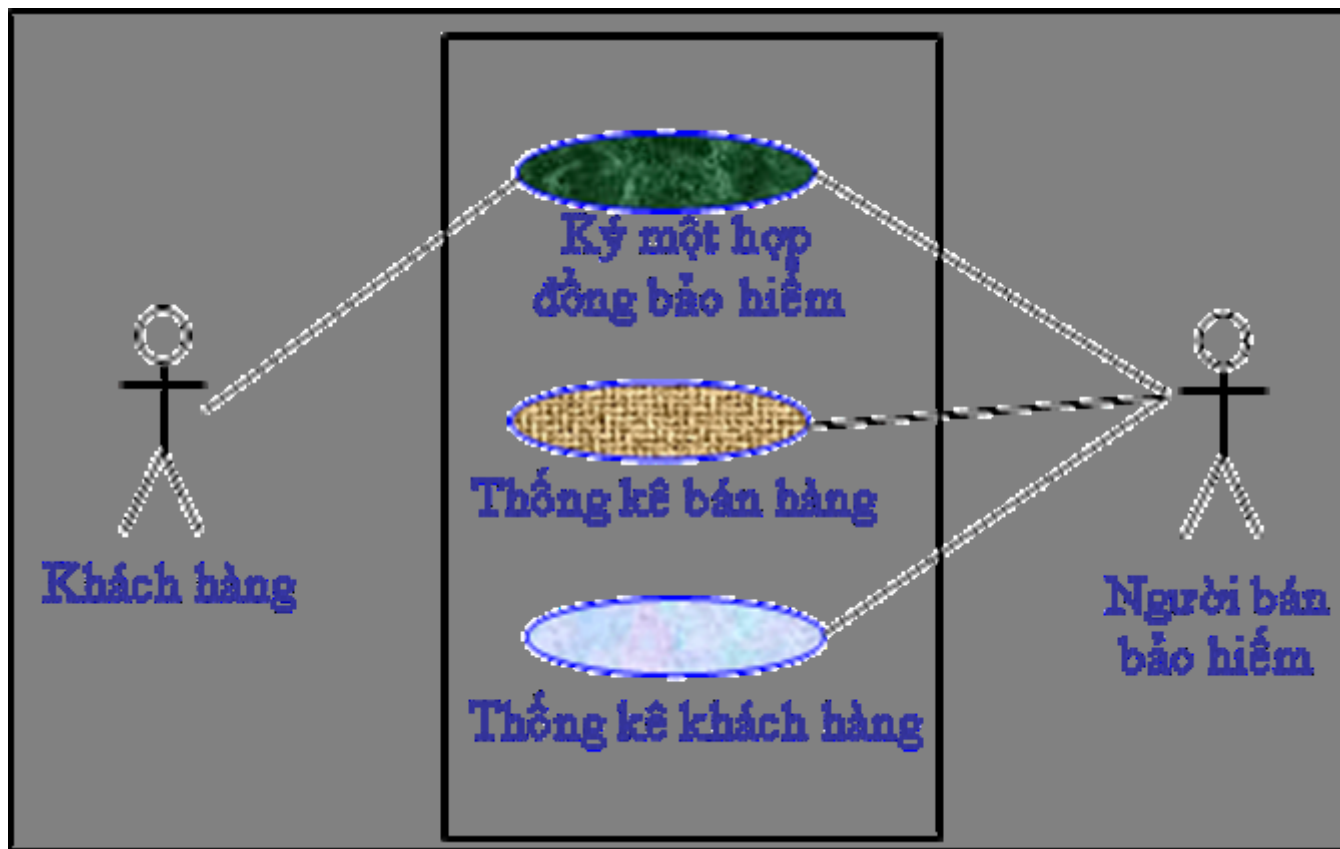


6.3.3.2. Ngôn ngữ mô hình hóa thống nhất UML (9)

- Biểu đồ Use case:
 - Use case (trường hợp sử dụng): là các chức năng mà các tác nhân bên ngoài đòi hỏi từ phía hệ thống.
 - Biểu đồ Use case chỉ ra số lượng các tác nhân ngoại cảnh và mối liên kết của các tác nhân đó với các chức năng mà hệ thống cung cấp.
 - Các Use case định nghĩa các yêu cầu về mặt chức năng đối với hệ thống.

6.3.3.2. Ngôn ngữ mô hình hóa thống nhất UML (10)

- Biểu đồ Use case của một công ty bán bảo hiểm:



6.3.4. *Đặc tả công nghệ dùng trong thiết kế*

- Phải nắm được các công nghệ sẵn có hiện tại.
- Lựa chọn công nghệ phù hợp với mục đích.
- Cần theo kịp xu hướng phát triển của công nghệ.

6.3.5. *Đánh giá và kiểm thử*

- Chuẩn bị và thực hiện các kiểm thử về tính tiện dụng của sản phẩm.
- Báo cáo kết quả kiểm thử cho nhóm thiết kế.

Tương tác người - máy (Human - Computer Interaction)

Giới thiệu môn học

- Giảng viên: TS. Bùi Thế Duy – Bộ môn Mạng & TTMT, phòng 304, nhà E3
- Email: duybt@vnu.edu.vn
- Giờ học: Chiều thứ 2, thứ 4 hàng tuần
2pm-5.20pm, phòng 210, 14C
- Website môn học:
<http://www.fotech.vnu.edu.vn/courses>
Chọn môn: HCI

Mục tiêu môn học

- Xây dựng khả năng thiết kế tốt cho những hệ thống tương tác ở các mức kỹ thuật, tính năng và nhận thức (cognitive) thông qua sự hiểu biết về các thách thức đang đối mặt với những người dùng của một hệ thống;
- Thu được một quy trình làm việc hợp lý để thiết kế giao diện;
- Khám phá và yêu thích môn học Tương Tác Người – Máy.

Giới thiệu chung

- “Tương tác người – máy” (HCI) là gì?
- Những chuyên ngành liên quan đến HCI
- Tại sao một học sinh CNTT phải quan tâm đến HCI
- Tầm quan trọng của một việc thiết kế tốt một giao diện người dùng

“Tương tác người – máy” (HCI) là gì?

- *HCI là sự nghiên cứu và phát triển các giao diện máy tính với mục đích làm cho người dùng dễ sử dụng chúng hơn.*
- HCI: tương tác người máy, giao tiếp người máy
KHÔNG CHỈ LÀ: thiết kế giao diện !!!!!
- HCI liên quan đến
 - Nghiên cứu việc con người sử dụng các giao diện
 - Phát triển các ứng dụng mới cho người dùng
 - Phát triển các thiết bị và công cụ mới cho người dùng

Những chuyên ngành liên quan đến HCI

- HCI liên quan đến nhiều ngành
- HCI nghiên cứu 3 phần:
 - Hình thức: các hình thức giao tiếp giữa người và máy
 - Chức năng: các chức năng mới trong giao tiếp người máy.
 - Cài đặt: Cài đặt các giao diện

Những chuyên ngành liên quan đến HCI

- Tâm lý học, xã hội học, triết học: chức năng
- Sinh lý học, Công thái học (Ergonomics): chức năng, hình thức

Công thái học (Ergonomics): Khoa học về việc thiết kế các máy móc, các công cụ, các máy tính và khu vực làm việc vật lý, sao cho mọi người dễ tìm thấy chúng và thoải mái trong sử dụng.

Những chuyên ngành liên quan đến HCI

- Thiết kế đồ họa và công nghiệp, thiết kế âm thanh, điện ảnh: hình thức, chức năng và cài đặt
- Kỹ nghệ phần mềm: chức năng và cài đặt
- Kỹ thuật điện, điện tử: cài đặt
- và có thể các ngành khác ...

Tại sao một học sinh CNTT phải quan tâm đến HCI

- Học sinh CNTT học để:
 - Phát triển các ứng dụng phần mềm mới
 - Phát triển các công cụ để dùng trong các ứng dụng: đồ họa 3 chiều, ngôn ngữ lập trình ...
 - Phát triển các hệ điều hành
 - ...

Con người là một trong những thành phần thiết yếu của các hệ thống này !!!!!

Tầm quan trọng của một việc thiết kế tốt một giao diện người dùng

- Giảm thời gian lập trình
- Giảm chi phí cho những trục trặc do giao diện
- Tăng khả năng bán được của sản phẩm
- Tăng năng suất
- Giảm những bệnh nghề nghiệp do dùng máy tính
- Giảm những lỗi nguy hiểm đến tính mạng

Tầm quan trọng của một việc thiết kế tốt một giao diện người dùng

- Trong lập trình:

- Một phần lớn mã liên quan đến giao diện
- Nếu thiết kế giao diện sai => phải làm lại
- Nếu không sửa được => người dùng phải sử dụng giao diện không tốt

Thiết kế giao diện tốt => giảm thời gian lập trình

Tầm quan trọng của một việc thiết kế tốt một giao diện người dùng

- Về mặt kinh tế:

- Tăng năng suất lao động
- Giảm chi phí đào tạo
- Giảm những lỗi người dùng
- Người sử dụng hài lòng
- Tạo ra những sản phẩm có chất lượng cao hơn

Tầm quan trọng của một việc thiết kế tốt một giao diện người dùng

- Tăng năng suất lao động

20 người dùng

x 230 ngày

x 100 màn hình giao tiếp 1 ngày

x 10 giây mỗi màn hình giao tiếp

= 1278 giờ (32 tuần)

Tầm quan trọng của một việc thiết kế tốt một giao diện người dùng

- Tăng năng suất lao động

5 người điều hành

x 500 lần chọn bảng một ngày

x 3 giây một lần chọn

x 230 ngày một năm

= 480 giờ (12 tuần)

Tầm quan trọng của một việc thiết kế tốt một giao diện người dùng

- Giảm chi phí đào tạo

20 nhân viên

x 2 ứng dụng mỗi năm

x 2.5 ngày mỗi ứng dụng

= 100 ngày (20 tuần)

Tầm quan trọng của một việc thiết kế tốt một giao diện người dùng

- Giảm những lỗi người dùng

500 người dùng
x 20 lỗi một năm
x 15 phút cho một lỗi

= mất 2500 giờ (63 tuần)

Tầm quan trọng của một việc thiết kế tốt một giao diện người dùng

- Tạo ra những sản phẩm có chất lượng cao hơn
 - Người dùng tiết kiệm thời gian khi sử dụng giao diện nên có thể tập trung vào công việc chính
 - Ví dụ: tìm kiếm dữ liệu, định dạng văn bản

Tầm quan trọng của một việc thiết kế tốt một giao diện người dùng

- Giảm những lỗi nguy hiểm đến tính mạng
 - Một chiếc máy bay đâm vào vách núi ở Colombia năm 1996 giết chết tất cả mọi người trên máy bay.
 - Lý do: người lái gõ phím “R” thay vì tên đầy đủ của sân bay. Hệ thống dẫn đường lấy ra trong hệ thống sân bay đầu tiên bắt đầu bằng chữ “R” – sai sân bay.
 - Kết quả: máy bay đâm vào núi
 - HCI có thể cứu sống tính mạng con người !!!

Tầm quan trọng của một việc thiết kế tốt một giao diện người dùng

- Tăng khả năng bán được của sản phẩm
 - DOS không thể so sánh được với các hệ điều hành khác cùng thời
 - Windows và Explorer đem lại cho Microsoft lợi nhuận cực lớn
 - Windows được sao chép lại từ giao diện của Macintosh !!!
 - Giao diện của Macintosh được sao chép lại từ Bravo – phát triển tại Xerox PARC !!!
- Giao diện đẹp dễ nhận được hợp đồng
- Giao diện tồi có thể bị loại ngay từ đầu cho dù chương trình tốt đến mấy

Tầm quan trọng của một việc thiết kế tốt một giao diện người dùng

- Máy tính đã xuất hiện khắp mọi nơi: điều khiển máy bay, ô tô, dàn nghe nhạc
- Giao diện người – máy tính tốt => giao diện người – các thiết bị tốt

Tương tác người - máy (Human - Computer Interaction)

Tính tiện lợi của một hệ thống

- Những yếu tố nào quyết định tính tiện lợi của một hệ thống ?

Ai có thể giúp tôi ????????????????

Tính tiện lợi của một hệ thống

- Phân tích ví dụ: Chiếc đàn violin
 - Cái gì làm cho nó tiện lợi ?
 -
 - hay không tiện lợi ?

Chiếc đàn violin có tiện lợi không?

- Người ta phải học 10 năm để có thể chơi đàn violin tốt
 - Khi bạn bắt đầu học – không thể tạo nổi một tiếng kêu
 - Hoặc có thể sau một hồi tập luyện
 - Nhưng chắc chắn là tiếng kêu sẽ cực kỳ kinh khủng !!!
- Vậy chiếc đàn violin có tiện lợi hay không?

Chiếc đàn violin có tiện lợi không?

- Tại sao chiếc đàn violin lại được tạo ra như vậy?
 - Vì nó được thiết kế cho các siêu nhân
 - Khi bạn làm chủ được nó, nó sẽ trở nên tiện lợi !!!
- Vậy một hệ thống có nên thiết kế như chiếc đàn violin
 - Có bao nhiêu người là siêu nhân??
 - Chắc cùng lắm là 10%
 - Vậy chúng ta không nên coi chiếc đàn violin là tiện lợi

Các nguyên tắc của tính tiện lợi

- Bốn nguyên tắc (Shackel, 1990): Tính có thể học được (learnability), tính dễ sử dụng (ease of use), tính linh động (flexibility) và tính cảm xúc (affectiveness)

B. Shackel. Human Factors and Usability. In: Preece, J. and L. Keller (Ed.), *Human-Computer Interaction: Selected Readings*. Prentice Hall, 1990.

Các nguyên tắc của tính tiện lợi

- ***Tính có thể học được***
 - các hệ thống tương tác phải dễ học
- ***Tính dễ sử dụng***
 - các hệ thống tương tác phải hiệu quả trong việc giúp người sử dụng đạt được mục đích
- ***Tính linh động***
 - các hệ thống tương tác phải thích ứng được những hoàn cảnh khác nhau
- ***Tính cảm xúc***
 - các hệ thống tương tác phải làm cho người dùng cảm thấy thoải mái

Các nguyên tắc của tính tiện lợi

- Tại sao phải theo các nguyên tắc này?
- Những nguyên tắc này giúp cho bạn tập trung vào mục tiêu đặt ra
- Chúng giúp cho bạn có thể đo được mức độ mục tiêu đặt ra được hoàn thành được đến đâu.
- Đây là một phương pháp đã được dùng để đánh giá một thiết kế !!!!!

Các nguyên tắc của tính tiện lợi

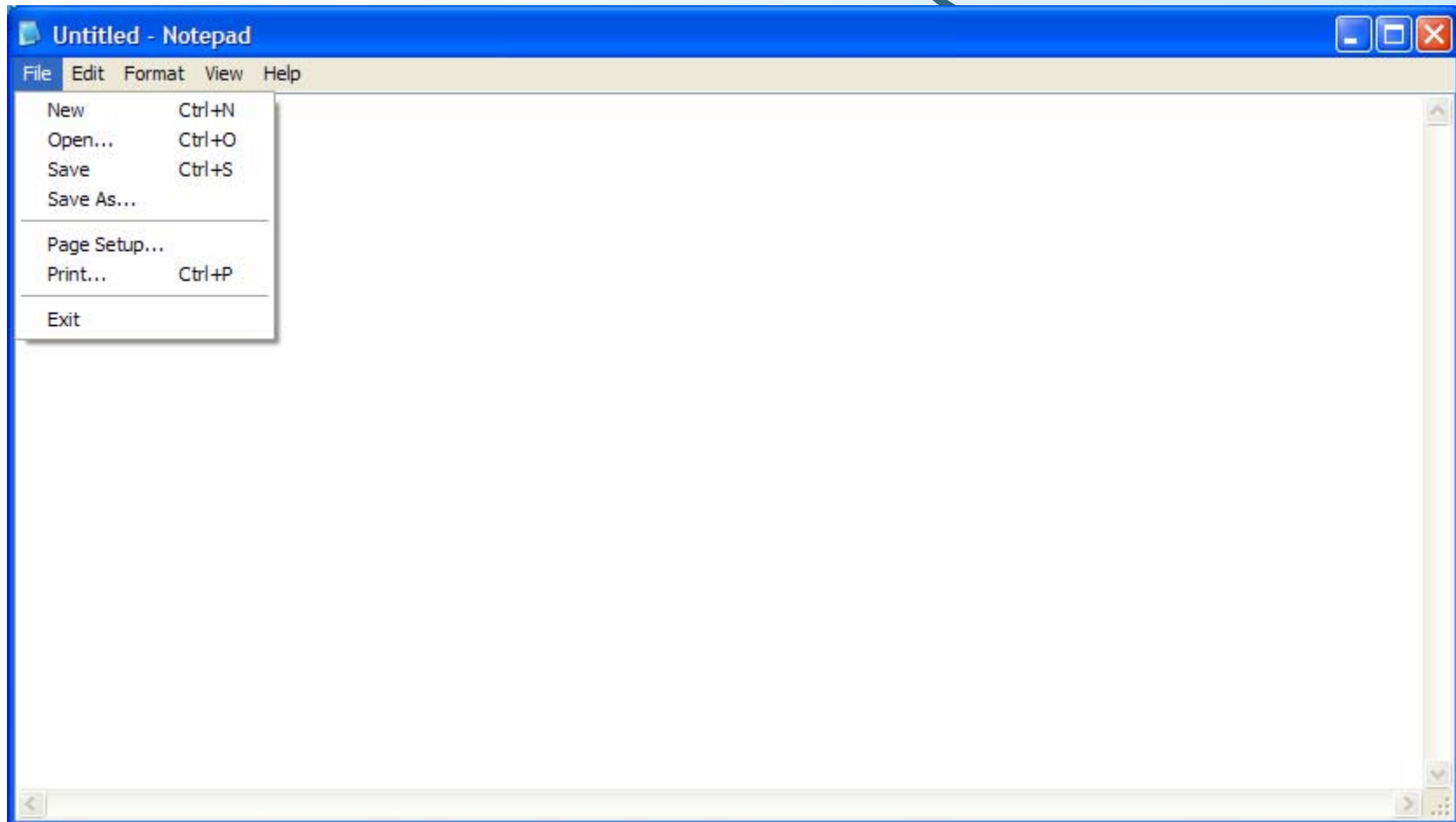
- ***Tính có thể học được***
- ***Tính dễ sử dụng***
- ***Tính linh động***
- ***Tính cảm xúc***

Tính có thể học được

- Tính có thể học được thể hiện qua thời gian và công sức bỏ ra để có thể đạt được một trình độ sử dụng nhất định.
- **Tính có thể tổng hợp được:** Khi tôi làm thế này thì cái gì đó xảy ra!
 - Hệ thống cho phép người dùng tổng hợp các kiến thức mà họ thu được thông qua việc sử dụng hệ thống từ đó có thể xây dựng một hệ thống các nguyên tắc về các kiến thức này.
- **Tính có thể đoán được:** Khi tôi làm thế này thì điều đó sẽ xảy ra!
 - Người dùng thường dự đoán kết quả của một sự tương tác dựa vào hệ thống kiến thức mà họ thu được từ các lần tương tác trước. Hệ thống nên hỗ trợ các suy luận hay dự đoán này bằng cách luôn luôn đưa ra các thông tin phản hồi nhất quán.

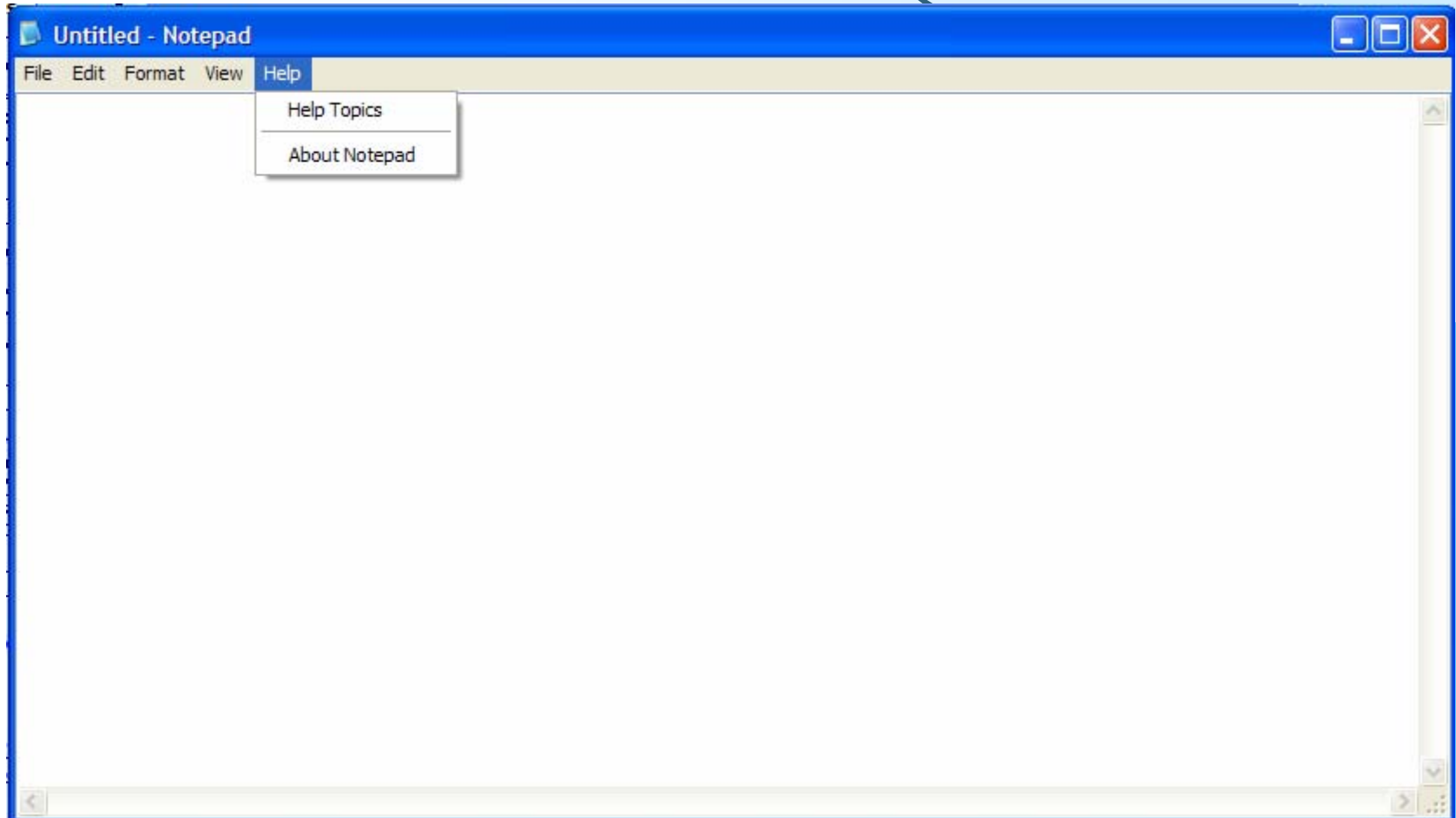
Tính có thể học được

Tính có thể tổng hợp được



Tính có thể học được

Tính có thể tổng hợp được



Tính có thể học được

Tính có thể đoán được

Mở tệp: cửa sổ hội thoại xuất hiện để chọn tên tệp.

=> Ghi tệp: chắc là một cửa sổ hội thoại cũng sẽ xuất hiện để chọn tên tệp

Tính có thể học được

- **Tính quen thuộc**

- Người dùng sử dụng các kiến thức có từ trước để quyết định xem sẽ phải làm gì trong các tình huống mới.
- Vận dụng các kiến thức theo quy luật, để người dùng đỡ phải suy nghĩ quá nhiều trước một tình huống mới.

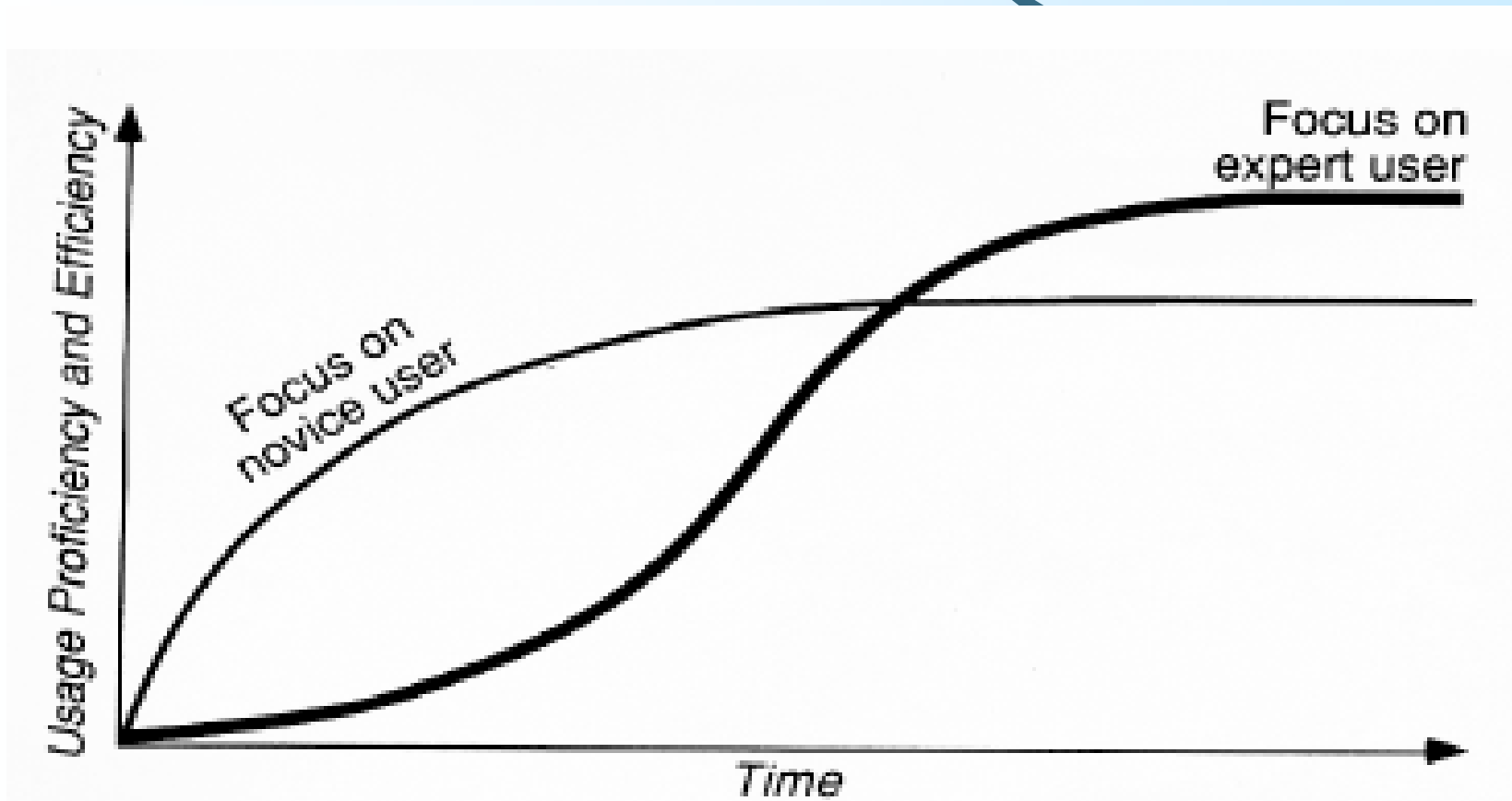
- **Tính khái quát và tính kiên định**

- Người sử dụng khái quát những quy luật và kỹ năng để áp dụng vào những tình huống tương tự.
- Hệ thống nên hỗ trợ việc tạo nên các kỹ năng bằng cách hoạt động tương tự trong những tình huống tương tự.

Tính có thể học được

- Làm thế nào để có thể đánh giá “tính có thể học được?”
 - Thông qua thời gian sử dụng để học được cách hoàn thành một công việc nhất định ở một trình độ nhất định

Tính có thể học được



Các nguyên tắc của tính tiện lợi

- *Tính có thể học được*
- ***Tính dễ sử dụng***
- *Tính linh động* ■
- *Tính cảm xúc*

Tính dễ sử dụng

- Tính dễ sử dụng được thể hiện qua mức độ dễ mà một nhiệm vụ có thể được hoàn thành một cách hiệu quả mà không có lỗi.
- **Tính quan sát được:** Tôi biết tôi đang làm gì!
 - Cho phép người dùng trực tiếp nhận biết được trạng thái của hệ thống.
 - Thiết kế hệ thống của bạn sao cho không chỉ hành động mà các trạng thái cũng nhìn thấy được
- **Tính phản ứng nhanh:**
 - Cho phép người dùng nhận biết được phản ứng cho hành động của họ ngay lập tức

Tính dễ sử dụng

- **Tính có thể khôi phục được:** Oops! Tôi vừa xóa mất toàn bộ ô cứng !!!
 - Người dùng có thể gây ra lỗi
 - Các hệ thống nên được thiết kế sao cho các lỗi có thể được ngăn chặn và khôi phục được.
- **Tính thích nghi với nhiệm vụ:** Các tính năng của hệ thống có thích nghi với nhiệm vụ của người dùng không?
 - Các hệ thống phải được thiết kế sao cho tính năng của chúng đáp ứng được nhiệm vụ của người dùng. Hay nói cách khác, các tính năng của hệ thống phải dựa trên nhiệm vụ của người dùng

Tính dễ sử dụng

- Làm thế nào để đánh giá được tính dễ sử dụng?
 - Mức hiệu suất công việc đạt được
 - Thời gian hoàn thành công việc ở mức cao nhất
 - Tần suất lỗi
 - Công sức đầu óc bỏ ra: thông qua câu hỏi điều tra hoặc các chỉ số sinh lý (v.d. nhịp tim)

Các nguyên tắc của tính tiện lợi

- *Tính có thể học được*
- *Tính dễ sử dụng*
- ***Tính linh động*** ■
- *Tính cảm xúc*

Tính linh động

- Tính linh động được thể hiện qua khả năng đáp ứng của một hệ thống đối với những người dùng khác nhau trong những trường hợp khác nhau.
- **Cửa sổ khởi tạo:**
 - Người sử dụng thích khởi tạo. V.d. : Windows Wizard.
- **Đa tuyến đoạn (Multithreading):**
 - Người dùng thường làm nhiều việc một lúc. Hệ thống nên được thiết kế để cho phép điều này.
- **Khả năng chuyển giao giữa các nhiệm vụ:**
 - Có khả năng phân biệt được người dùng nào đang làm gì. V.d.: Tự động sửa chính tả.

Tính linh động

- **Tính thay thế được:** Cho phép những giá trị vào ra tương đương có thể được thay thế thoải mái. V.d: kéo thả
- **Tính tùy biến (Customizability):** Cho phép một hệ thống có thể thích nghi được với nhu cầu của người dùng

Tính linh động

- Làm thế nào để đánh giá được tính linh động?
 - Thông qua hiệu suất và số lỗi tạo ra trong các tình huống khác nhau

Các nguyên tắc của tính tiện lợi

- *Tính có thể học được*
- *Tính dễ sử dụng*
- *Tính linh động*
- *Tính cảm xúc*

Tính cảm xúc

- Tính cảm xúc thể hiện qua mức độ ưa thích và thoải mái người sử dụng cảm thấy khi sử dụng một hệ thống.
-
- **Giá trị cảm xúc: !!!!**
- **Tính thẩm mỹ**
- **Các yếu tố môi trường:** Các hệ thống nên được thiết kế để phù hợp với các môi trường vật lý: RSI (Repetitive Strain Injury)!

Tính cảm xúc

- Làm thế nào để có thể đánh giá được tính cảm xúc?
 - Thông qua câu hỏi điều tra hoặc các chỉ số sinh lý

Tương tác người - máy (Human - Computer Interaction)

Giới thiệu môn học

- Giảng viên: TS. Bùi Thế Duy – Bộ môn Mạng & TTMT, phòng 304, nhà E3
- Email: duybt@vnu.edu.vn
- Giờ học: Chiều thứ 2, thứ 5 hàng tuần
2pm-5.20pm, phòng 5.1, nhà E4
- Website môn học:
<http://www.fotech.vnu.edu.vn/courses>
Chọn môn: HCI

Con người

- Nhân vật trung tâm trong mọi hệ thống tương tác
- Máy tính được thiết kế để phục vụ con người => Yêu cầu của người là ưu tiên số một.
- Trong phần này chúng ta quan tâm đến các yếu tố về con người – một vấn đề tưởng chừng như chẳng liên quan gì đến HCI

Con người

- Để có thể phục vụ con người được tốt:
 - chúng ta cần phải biết một người nào đó có khả năng làm gì và không làm được gì V.d. : người khiếm thị: dùng âm thanh, phím nổi, chữ nổi
 - chúng ta cần biết con người nhận biết thế giới ntn, lưu trữ, xử lý thông tin và giải quyết vấn đề ntn, và thao tác với các đồ vật ntn

Con người

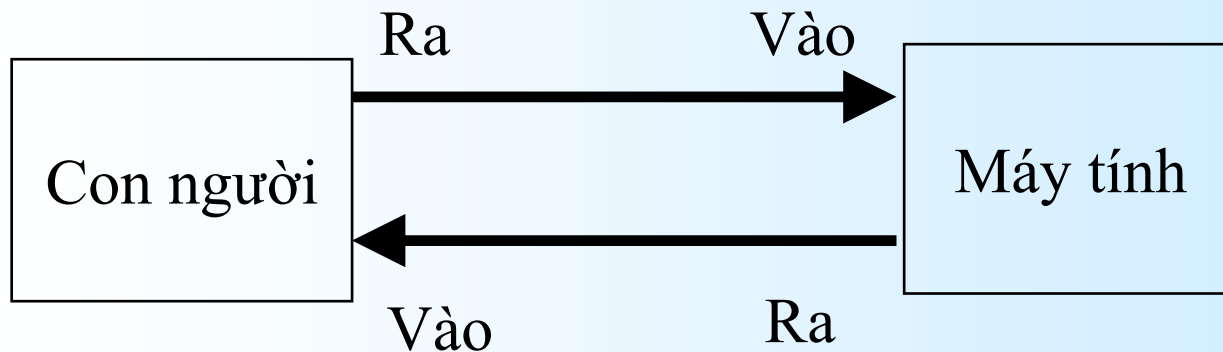
- Chúng ta sẽ chỉ quan tâm đến những đặc điểm của con người liên quan đến HCI !
- Để làm được điều đó, chúng ta sẽ nghiên cứu con người ở một mô hình đơn giản hơn – như một hệ thống xử lý thông tin
- Mô hình đó có 3 thành phần chính:
 - Thành phần vào/ra
 - Bộ nhớ
 - Bộ xử lý

Con người

- Mô hình hóa con người:
 - Thành phần vào/ra
 - Bộ nhớ
 - Bộ xử lý

Thành phần vào ra của con người

- Con người giao tiếp với thế giới thông qua nhận và gửi thông tin bằng các thành phần vào ra



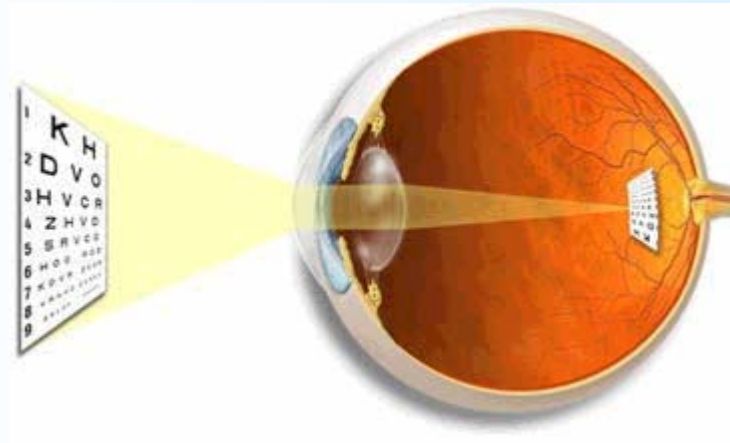
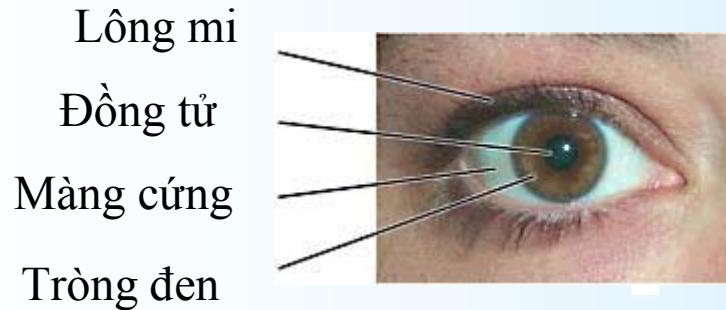
Thành phần vào

- Thông qua năm giác quan: thị giác, thính giác, khứu giác, vị giác, xúc giác
- Hiện nay, ba giác quan là quan trọng cho HCI, hai giác quan vị giác và khứu giác vẫn chưa được quan tâm đến

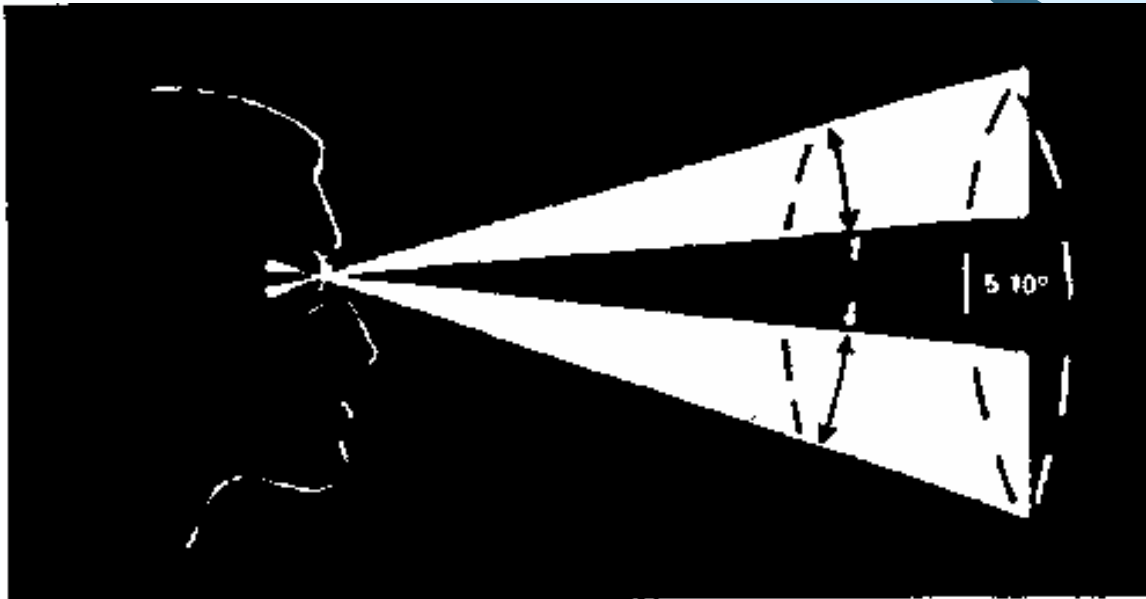
Thành phần vào

- **Thị giác**
- Thính giác
- Xúc giác

Thành phần vào – thị giác



Thành phần vào – thị giác

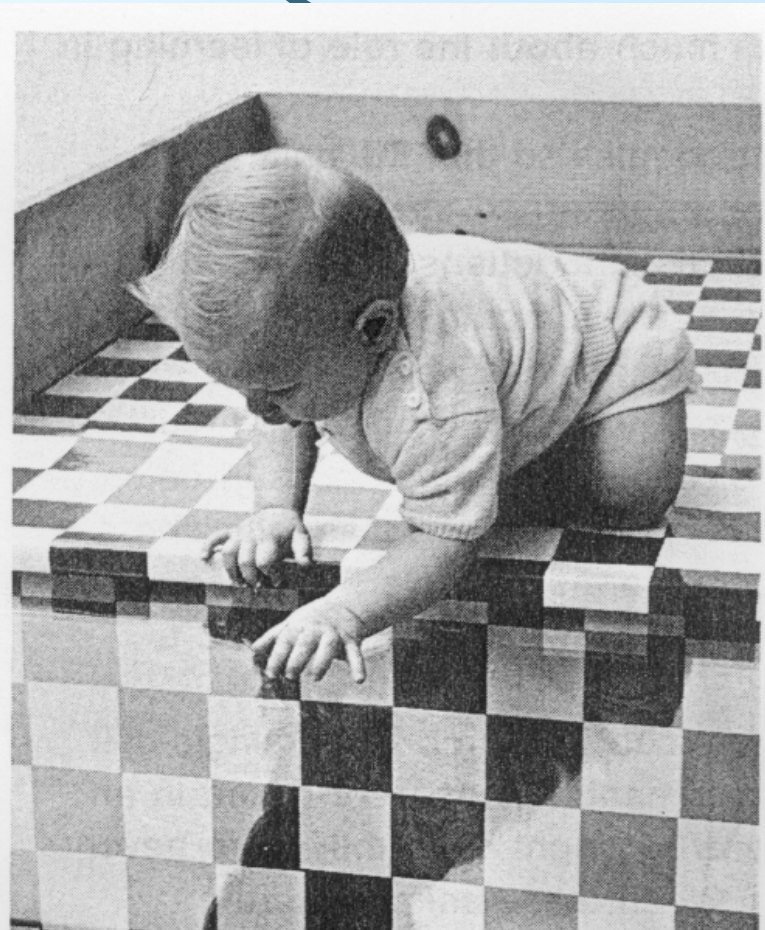


Điểm mù buổi tối

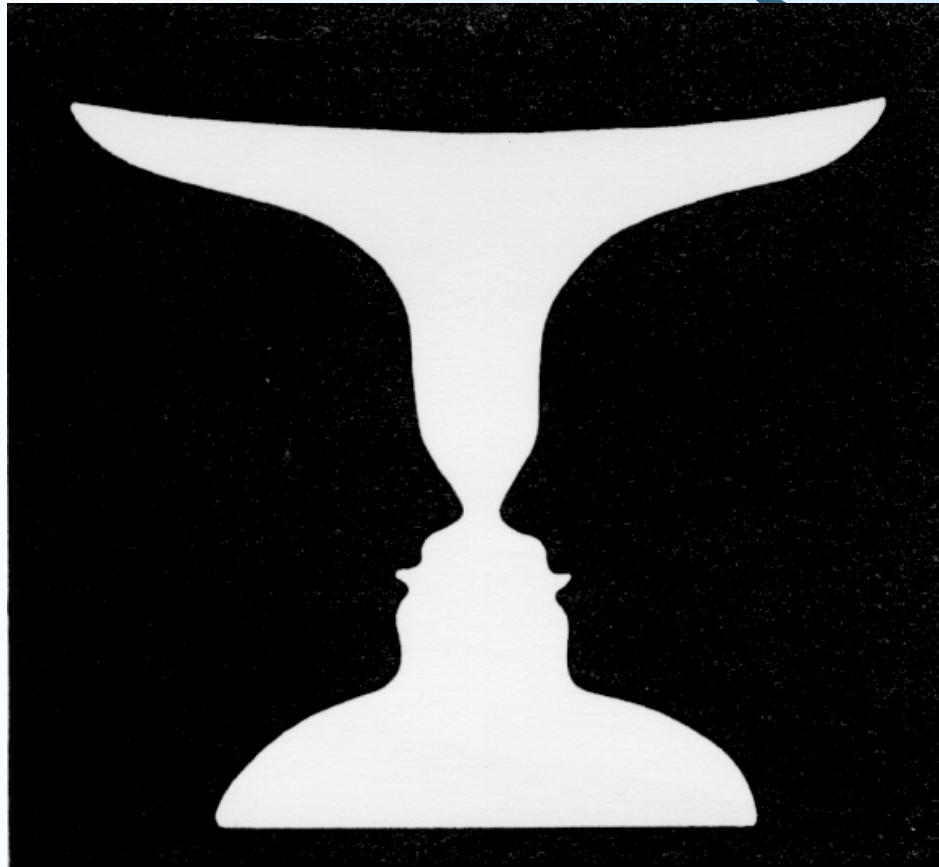
Cách bố trí màu nổi (foreground) và màu nền (background)

foreground	Back ground							
	black	blue	brown	cyan	green	magenta	red	white
black	--			+		+		++
blue	-	--						++
cyan	++			■	--	-		-
green	++	+		+	--			
red	+	--	-		-	--	--	+
white	+	++						--
yellow	++	++		+		+		
deep green	+				--			
deep cyan	+	+		+	-	+	+	-
deep magenta	+				++	-	-	
deep white	++			+	+			--

Vách ảo



Đảo hình và nền – chiếc ly



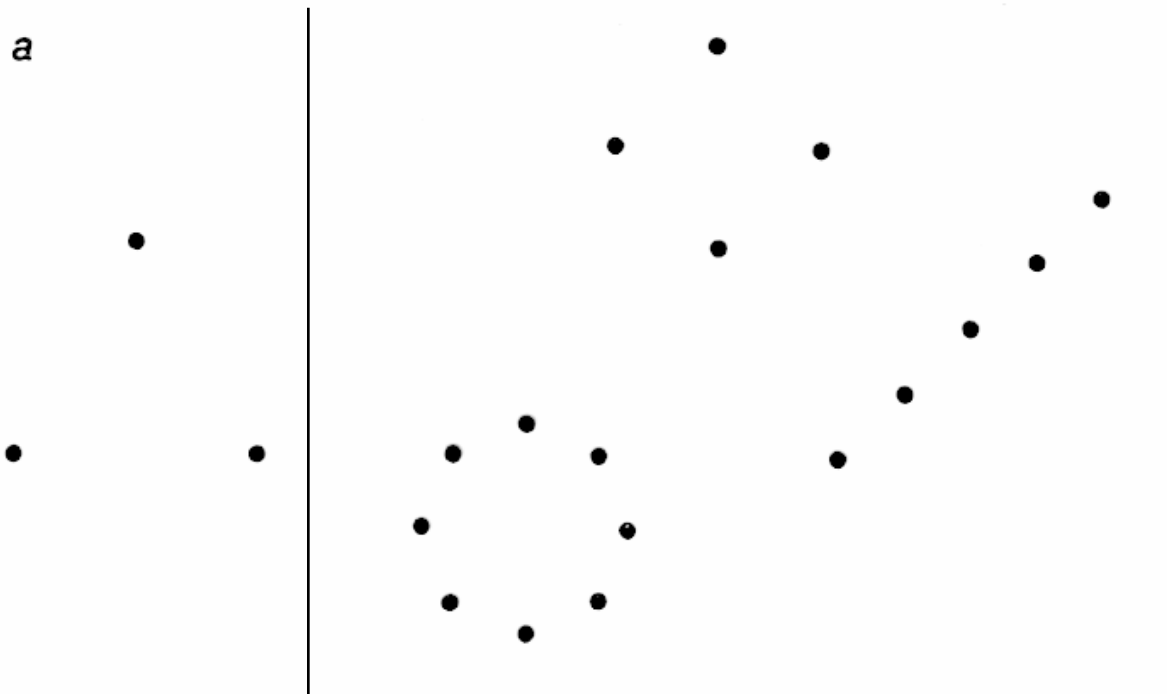
Cảm nhận hình ảnh

- Hình ảnh không chỉ là được cảm nhận một cách thụ động mà còn theo một số luật cơ bản để nhận được “hình đẹp”:
 - Luật Praegnantz: Trong số một số tổ chức hình ảnh được xuất hiện đồng thời, tổ chức hình ảnh nào đơn giản nhất, ổn định nhất sẽ được cảm nhận.
 - Luật kề cận: hướng về nhóm những thành phần ở cạnh nhau

Cảm nhận hình ảnh

- Luật tương tự: hướng về nhóm những thành phần tương tự nhau
- Luật liên tục: hướng về nhóm những thành phần liên tiếp nhau tạo thành các đường cong mượt
- Luật đóng: hướng về nhóm những thành phần tạo thành một hình đóng

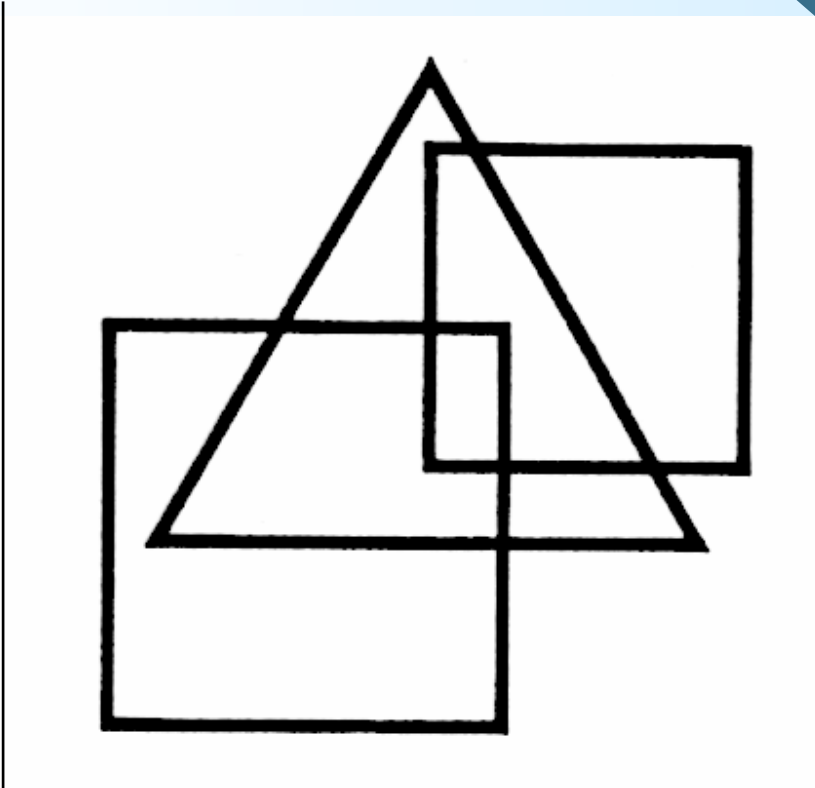
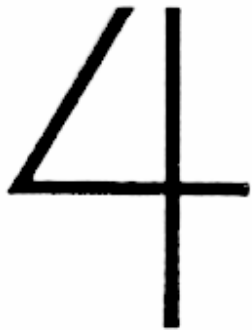
Hình ảnh ẩn



Hình tam giác
bên trái được
giấu ở hình bên
phải

Hình ảnh ẩn

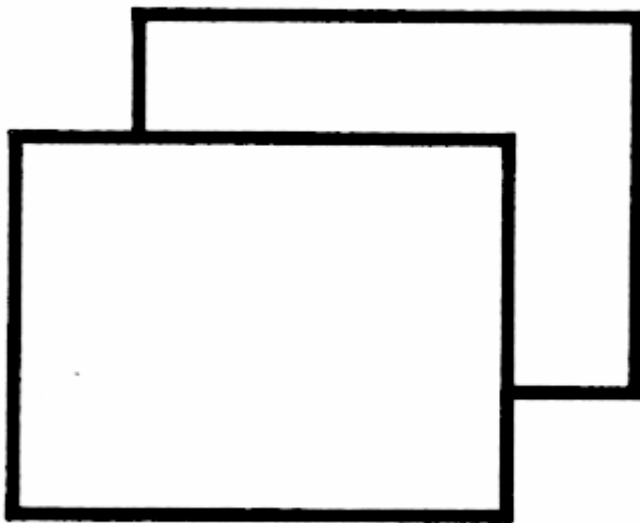
b



Số 4 bên trái
được giấu ở
hình bên phải

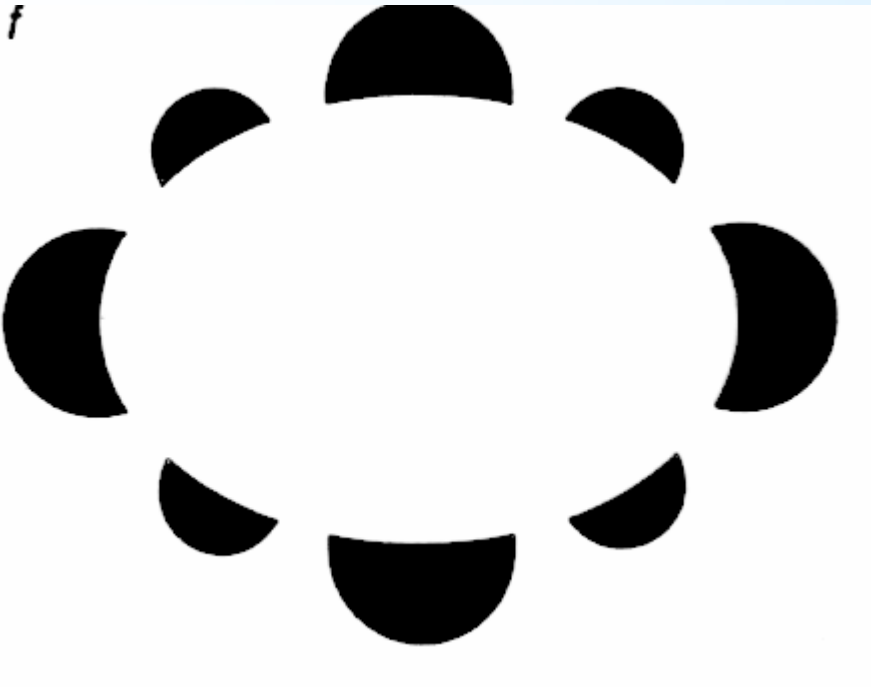
Cảm nhận hình ảnh

e



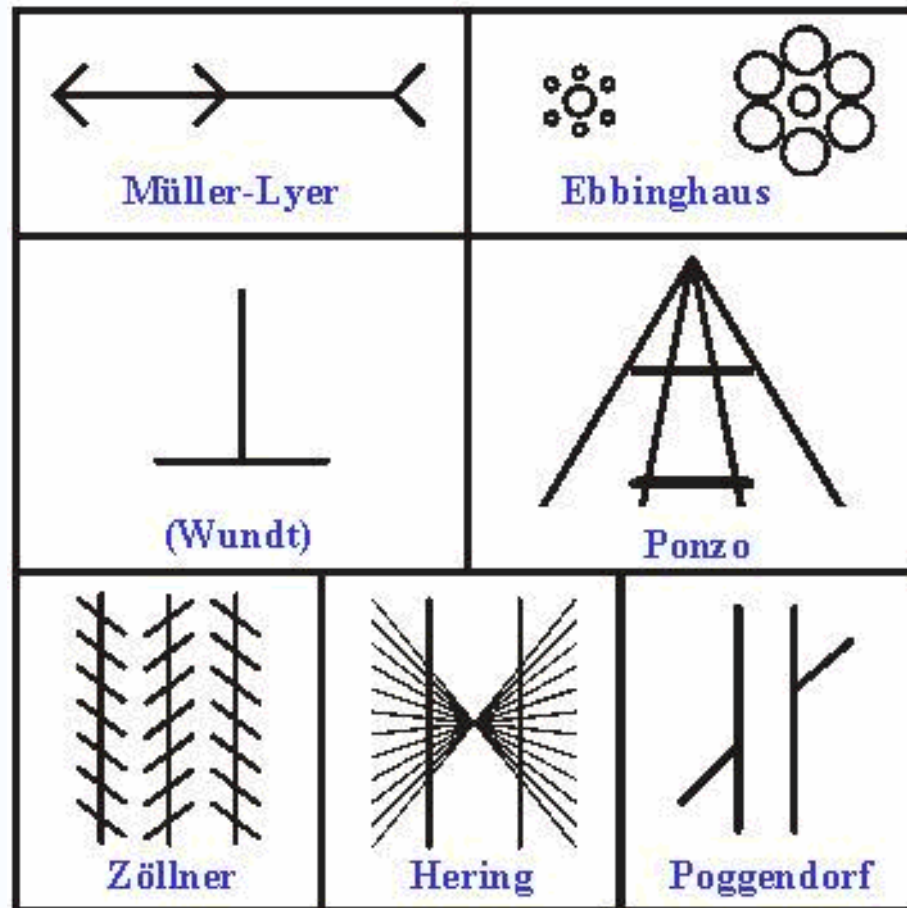
Chúng ta cảm nhận được rằng có hai hình chữ nhật đè lên nhau

Cảm nhận hình ảnh

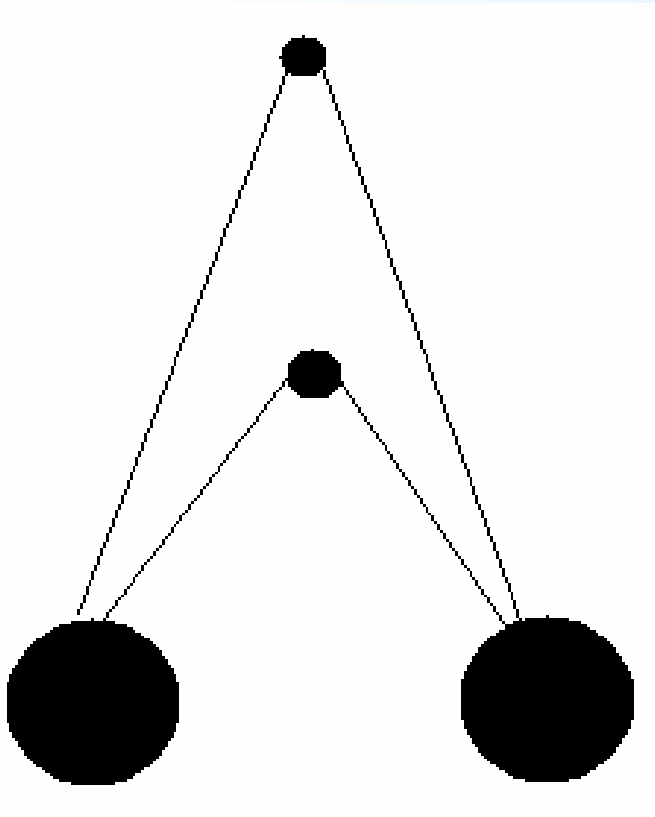


Chúng ta vẫn cảm nhận được một hình oval mặc dù hình này không được vẽ rõ ràng

Những ảo giác



Cảm nhận về độ sâu



- Thông qua hai mắt

Cảm nhận về độ sâu - thông qua chi tiết của hình ảnh

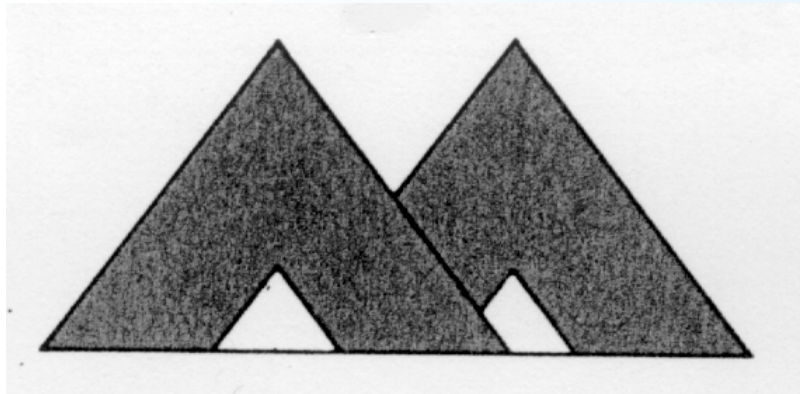


Cảm nhận về độ sâu - thông qua chuyển động

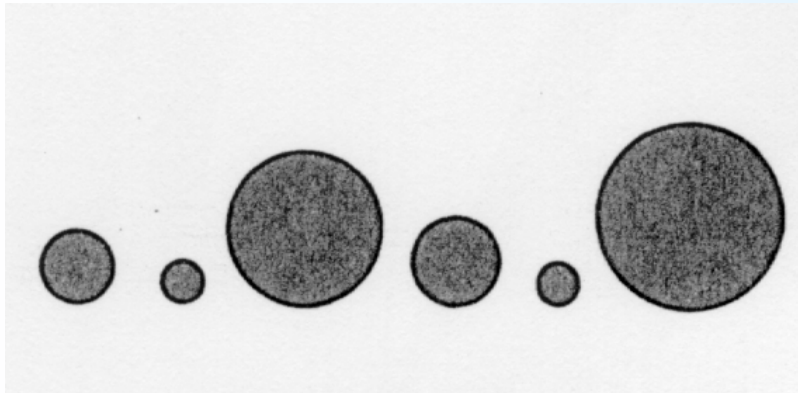


Vật ở gần chuyển động nhanh hơn, vật ở xa chuyển động chậm hơn

Cảm nhận về độ sâu



Theo vị trí



Theo kích cỡ

Ứng dụng Màn hình 3D – phim 3D



Thu hai hình ảnh dành cho 2 mắt (camera có len đôi)

Hai hình ảnh được chiếu đồng thời cho 2 mắt

Dùng kính để xem 2 hình ảnh ở hai mắt khác nhau

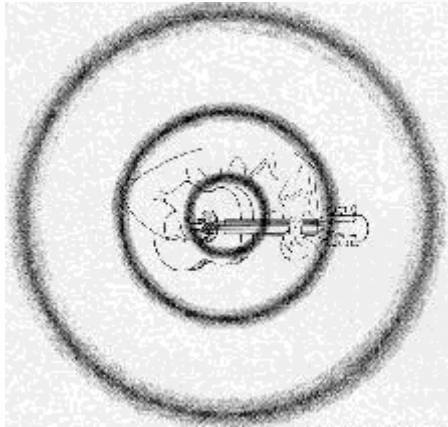
Thành phần vào

- Thị giác
- **Thính giác**
- Xúc giác

Thành phần vào – thính giác

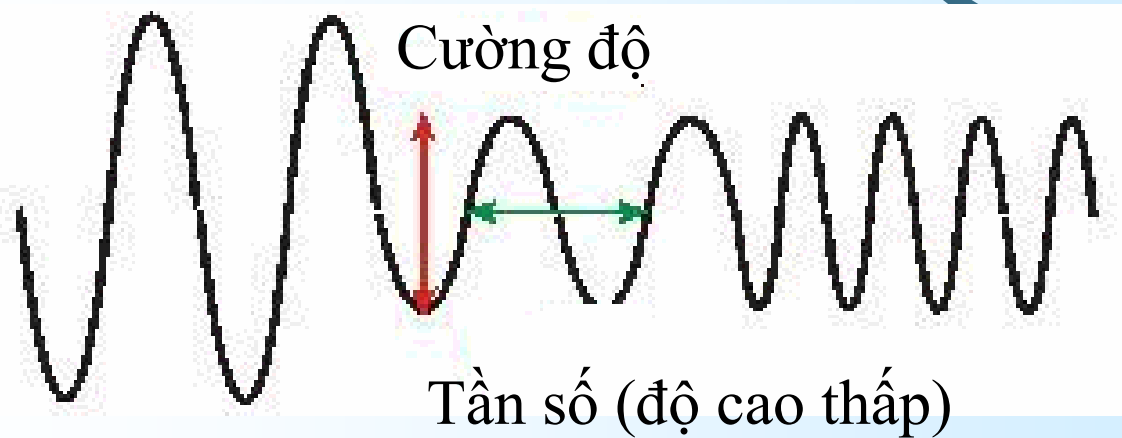
- Thính giác quan trọng với chúng ta như thế nào:
 - Hệ thống báo động
 - Cân bằng cảm xúc: nghe nhạc
 - Giao tiếp
- Những vấn đề thú vị nhất về thính giác
 - Cơ bản của hòa âm
 - Sự ảnh hưởng của kiến thức tới việc tách các nguồn âm thanh khác nhau
 - Nhận dạng các giọng nói

Bản chất của âm thanh



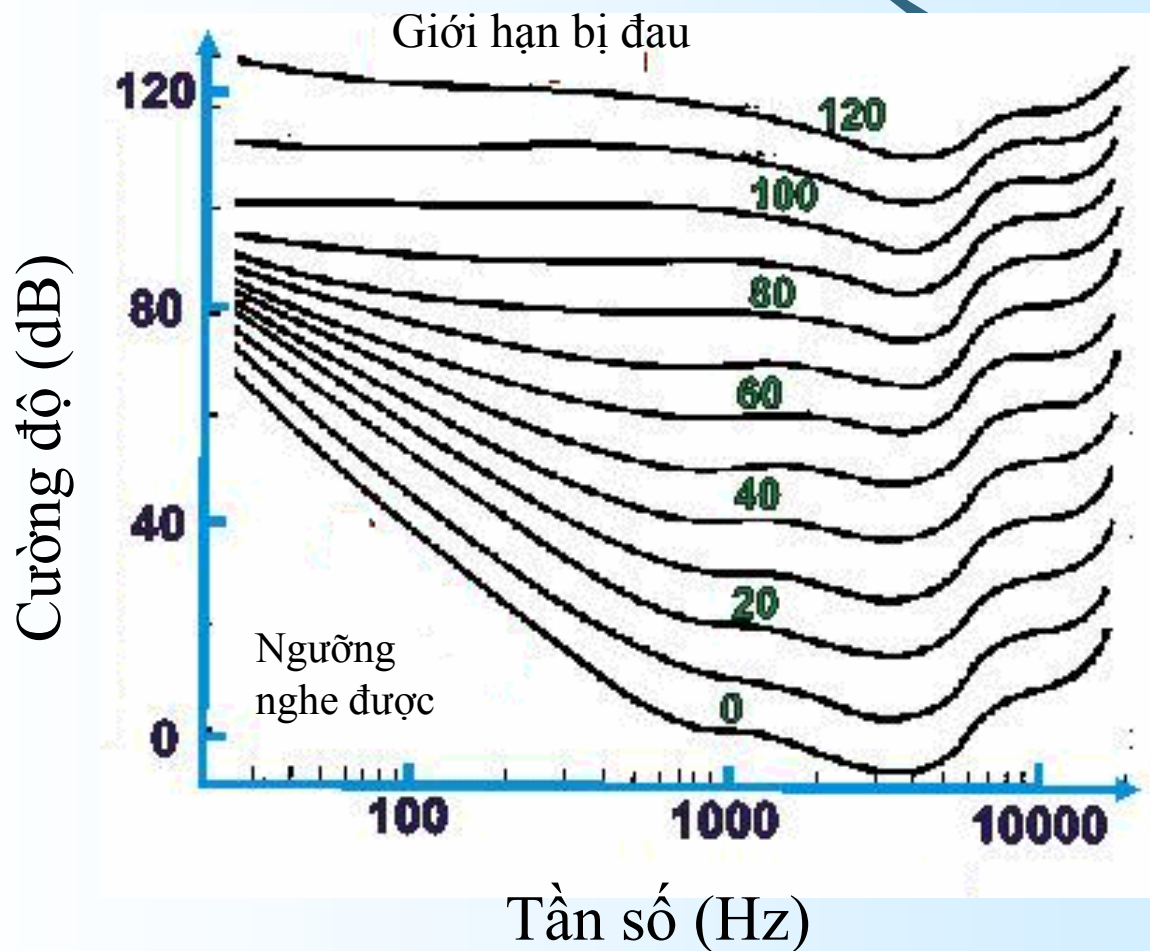
- Nguồn âm thanh phát ra các sóng tròn
- “Tiếng ồn trắng” (white noise) là sự tổng hợp của rất nhiều âm thanh với tần số và cường độ ngẫu nhiên (tiếng gợn sóng trên hồ)

Bản chất của âm thanh



Một âm thanh trong trẻo được thể hiện bằng dạng sóng hình sin (không có nhiễu) với biên độ là cường độ và tần số là độ cao thấp

Cảm nhận âm thanh



Cảm nhận âm thanh

- Con người chúng ta có thể cảm nhận được:
 - tần số của âm thanh (độ cao thấp)
 - cường độ (độ to nhỏ)
 - âm sắc (khác về nguồn âm thanh – phát ra từ các dụng cụ khác nhau, mặc dù có cùng tần số và cường độ)
- Chúng ta cũng có thể nhận ra vị trí của nguồn âm thanh
 - dùng hai tai
 - sự khác nhau về thời gian nhận và cường độ

Cảm nhận âm thanh

- Con người có thể nghe được âm thanh với các từ số từ khoảng 20Hz tới 15kHz
- Con người có thể phân biệt được thay đổi khoảng 1.5Hz ở các tần số thấp. Càng tần số cao thì khả năng phân biệt càng kém hơn
- Những tần số khác nhau thì ảnh hưởng đến những phần khác nhau trong hệ thống xử lý âm thanh của con người => tạo ra những kích thích khác nhau đến hệ thống thần kinh

Cảm nhận âm thanh

- Hệ thống xử lý âm thanh thực hiện một số thao tác lọc âm thanh => cho phép bỏ qua những tiếng ồn nền => tập trung vào âm thanh quan trọng
- “Cocktail party effect” – chúng ta vẫn có thể nghe thấy ai đó gọi chúng ta trong một căn phòng rất ồn

HCI với thính giác

- Các âm thanh hiện vẫn đang được sử dụng chính vào việc thông báo:
 - thông báo khi gõ nhầm nút
 - thông báo khi vào Windows
 - thông báo khi máy sắp hết pin
- Hiện nay âm thanh đang được nghiên cứu:
 - Tổng hợp tiếng nói => nghe đọc tài liệu thay vì nhìn tài liệu => phục vụ người khiếm thị
 - Dùng âm nhạc để tạo ra các hiệu ứng trong trình diễn nội dung

Thành phần vào

- Thị giác
- Thính giác
- **Xúc giác**

Thành phần vào – xúc giác

- Xúc giác: khả năng phát hiện và hiểu được những thông tin cảm nhận được từ da
- Các nghiên cứu đã chỉ ra rằng khả năng cảm nhận bằng xúc giác của trẻ nhỏ có thể đã được phát triển trước khả năng phân biệt vật thể bằng thị giác
- Theo bạn, môn thể thao nào (không) cần xúc giác?

Ba loại cảm nhận bằng xúc giác

- sức ép:
 - mạnh
 - nhẹ
- nhiệt độ:
 - lạnh
 - ấm
- mức độ đau:
 - nhọn
 - tù

Cảm nhận bằng xúc giác với HCI

- V.d.: chúng ta cảm nhận được phím bị bấm xuống
-
- Các thiết bị mới: haptic

Thành phần ra

- Thông qua một số các cơ quan như tay, chân, mắt, đầu và hệ thống tiếng nói.
- Khi giao tiếp với máy tính, hiện nay tay vẫn đóng vai trò chính trong việc gõ bàn phím và điều khiển chuột
- Tiếng nói, ánh mắt và chuyển động của đầu đang được nghiên cứu để điều khiển máy tính

Tương tác người - máy (Human - Computer Interaction)

Con người

- Mô hình hóa con người:
 - Thành phần vào/ra
 - **Bộ nhớ**
 - Bộ xử lý

Bộ nhớ con người

- Bộ nhớ giác quan (sensory memory)
- Bộ nhớ ngắn hạn (short-term memory)
- Bộ nhớ dài hạn (long-term memory)

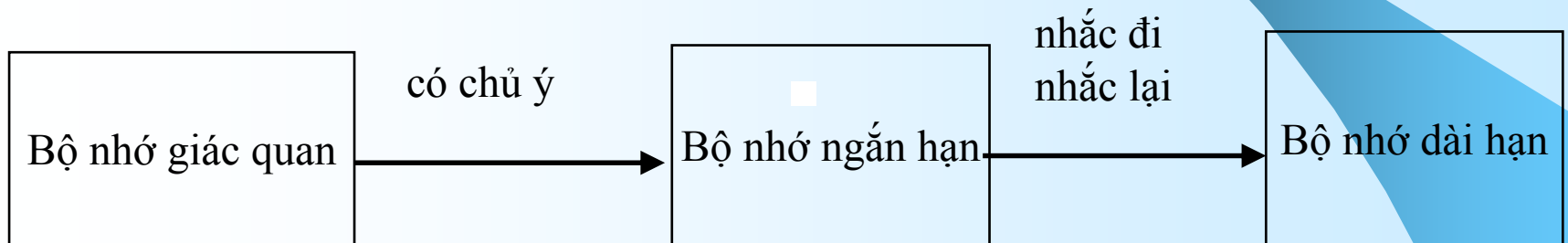
Bộ nhớ giác quan

- Vùng đệm chứa các tín hiệu nhận vào bằng các giác quan:
 - Bộ nhớ hình tượng (iconic memory) cho thị giác
 - Bộ nhớ tượng thanh (echoic memory) cho thính giác
 - Bộ nhớ xúc giác (haptic memory) cho xúc giác

Bộ nhớ giác quan

- Các bộ nhớ này liên tục bị ghi đè bởi những tín hiệu mới
- V.d.: hiện tượng lưu ảnh
- V.d.: tai lưu thông tin trong một thời gian ngắn – 2 tai nhận được một âm thanh tại hai thời điểm khác nhau (rất gần nhau) -> xác định được âm thanh được phát từ đâu

Bộ nhớ con người



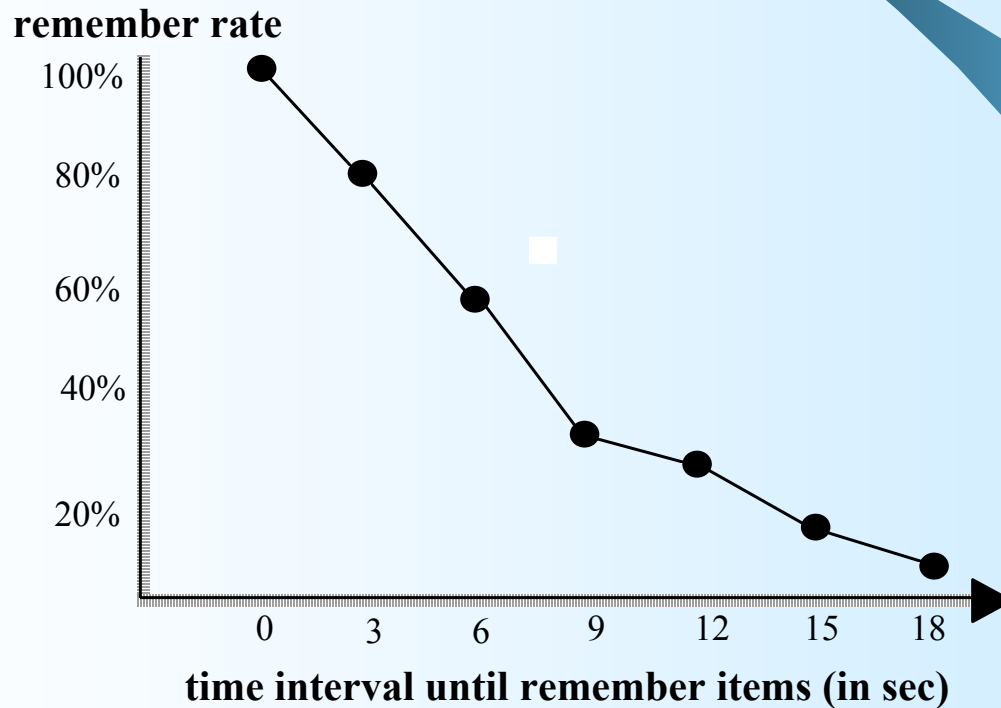
Bộ nhớ ngắn hạn

- Hay còn gọi là bộ nhớ làm việc
- V.d. để tính 35×6 , chúng ta có thể nhắm 30×6 rồi cộng với 5×6 hoặc 35×2 ra 70 rồi lấy 70×3
- V.d. đọc sách, chúng ta phải nhớ một số thông tin thì mới hiểu được quyển sách: các từ trong câu đang đọc, một số câu trước đó, một số chi tiết trước đó

Bộ nhớ ngắn hạn

- Con người có thể nhớ 7 ± 2 mục liên tiếp:
 - Bạn thử nhớ 2419406832 xem các bạn nhớ được bao nhiêu số?
 - Thế còn 764 321 5793 ?

Bộ nhớ ngắn hạn



Bộ nhớ dài hạn

- Dung lượng lớn hơn nhiều so với bộ nhớ ngắn hạn
 - Thời gian truy cập lâu hơn
 - Nhiều thông tin:
 - thông tin cũ nhiều thông tin mới được học
 - thông tin mới nhiều các thông tin cũ
- => Học nhiều quên nhiều, học ít quên ít???

Bộ nhớ dài hạn

- Có hai loại bộ nhớ dài hạn:
 - Loại nhớ theo tình tiết (episodic)
 - Loại nhớ theo ngữ nghĩa (semantic)

Bộ nhớ dài hạn

Loại nhớ theo tình tiết

- Bộ nhớ loại này ghi lại các sự kiện và kinh nghiệm theo cấu trúc chuỗi
-
- Giúp chúng ta nhớ lại các sự kiện đã xảy ra trong quá khứ

Bộ nhớ dài hạn

Loại nhớ theo ngữ nghĩa

- Bộ nhớ loại này ghi lại các khái niệm, sự thật và các kỹ năng chúng ta học được theo cấu trúc liên kết
- Các thông tin trong bộ nhớ loại này nhận được từ bộ nhớ theo tình tiết, cho phép chúng ta học được các khái niệm và sự thật mới từ kinh nghiệm

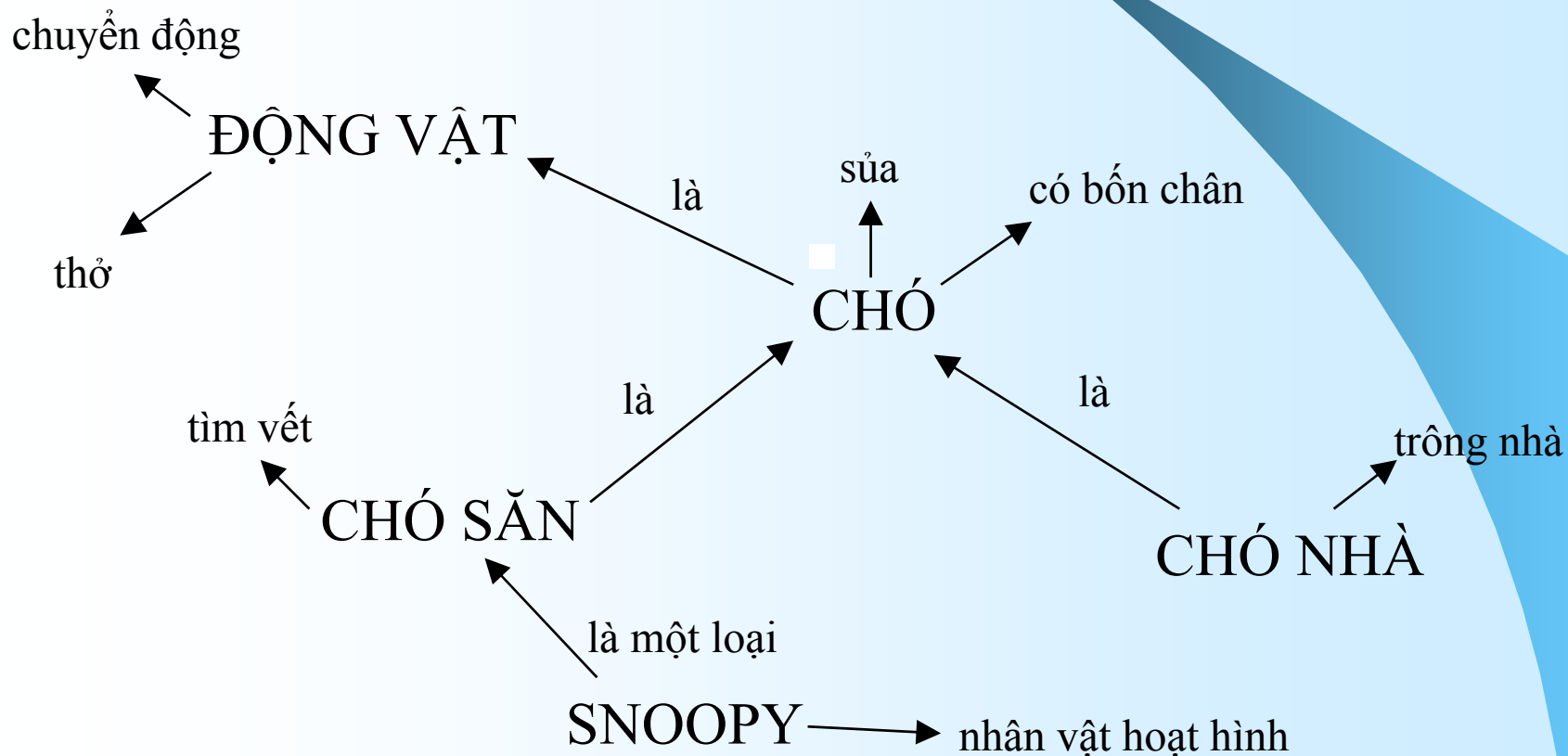
Bộ nhớ dài hạn

Loại nhớ theo ngữ nghĩa

- Bộ nhớ này được tổ chức để cho phép chúng ta truy cập thông tin, các mối quan hệ giữa các thông tin và cho phép chúng ta suy diễn
- Bộ nhớ này thường được biểu diễn dưới dạng mạng lưới – mạng lưới ngữ nghĩa (semantic network)

Bộ nhớ dài hạn

Loại nhớ theo ngữ nghĩa



Bộ nhớ dài hạn

Loại nhớ theo ngữ nghĩa

- Cho phép chúng ta suy diễn: vì chó có 4 chân => chó săn có 4 chân
- Lưu ý: có những liên kết nối sang hẳn những lĩnh vực khác (ví dụ như Snoopy -> phim hoạt hình)

Bộ nhớ dài hạn

Loại nhớ theo ngữ nghĩa

- Sự tồn tại của mạng ngữ nghĩa này trong con người chúng ta được chứng minh bởi Collins và Quillian (1969):
 - Một số người được hỏi về các thuộc tính của một số đối tượng. Thời gian suy nghĩ để trả lời được ghi lại. Kết quả là: người ta suy nghĩ lâu hơn khi được hỏi những câu hỏi kiểu như: “chó săn có thỏ không” so với các câu hỏi kiểu như: “chó săn có tìm vết được không?”
 - Lý do: con người phải tìm kiếm thông qua mạng ngữ nghĩa, suy ngược lên để tìm ra câu trả lời

A.M. Collins and M.R. Quillian. Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behaviour*, 8:240-247, 1969.

Bộ nhớ dài hạn

- Một số loại cấu trúc khác của bộ nhớ cũng được đưa ra để giải thích chúng ta lưu trữ kiến thức như thế nào:
 - Kiểu khung
 - Kiểu kịch bản
 - Kiểu quy tắc

Bộ nhớ dài hạn

Kiểu khung

CHÓ

Tính chất cố định: có 4 chân

Tùy biến:

- kích cỡ
- màu sắc

Bộ nhớ dài hạn

Kiểu kịch bản

Kịch bản khi đưa chó đi khám

Điều kiện:

- chó bị ốm
- bệnh viện thú y mở cửa
- người chủ có tiền

Kết quả:

- chó khỏi ốm
- người chủ nghèo hơn
- bác sĩ giàu hơn

Cảnh:

- đến phòng khám
- ngồi chờ
- bác sĩ khám
- trả tiền

Bộ nhớ dài hạn

Kiểu quy tắc

NẾU chó vẫy đuôi
THÌ lại gần

NẾU chó nhe răng
THÌ chạy

Bộ nhớ dài hạn – xử lý

- Ba hoạt động chính:
 - Ghi nhớ
 - Quên
 - Truy cập thông tin

Bộ nhớ dài hạn – Ghi nhớ

- Do quá trình nhắc đi nhắc lại từ bộ nhớ ngắn hạn
- Ebbinghaus (1885): tự thí nghiệm và rút ra kết luận: lượng thông tin nhớ được tỷ lệ với thời gian học

H. Ebbinghaus. *Über das Gedächtnis*. Dunker 1885. Translated by H. Ruyber and C. E. Bussenius, 1913, *Memory*, Teacher's College, Columbia University.

Bộ nhớ dài hạn – Ghi nhớ

- Baddeley và Longman (1978): Việc ghi nhớ hay học tập sẽ đạt hiệu quả tốt hơn nếu nó được dàn trải đều theo thời gian

=> Khóa học master của chúng ta: hơi vất vả

A. D. Baddeley and D. J. A. Longman. The influence of length and frequency of training sessions on rate of learning to type. *Ergonomics*. 21: 627-635, 1978.

Bộ nhớ dài hạn – Ghi nhớ

- Nhớ các từ mô tả các đối tượng dễ hơn các từ mô tả các khái niệm:
 - Chuỗi 1: Nhà - cửa - cây - mèo - chó - ô tô
 - Chuỗi 2: Tuổi tác - logic - lạnh - im lặng – quá khứ - chủ nghĩa
- Những thông tin có ý nghĩa và quen thuộc thì dễ nhớ hơn: đọc thần thoại Hy Lạp thì khó nhớ hơn thần thoại Việt Nam, châu Á

Bộ nhớ dài hạn – Quên

- Nếu quá trình ghi nhớ thông tin được trợ giúp bằng các cấu trúc, sự quen thuộc và tính cụ thể, làm thế nào để chúng ta có thể mất thông tin, có thể quên ?

Bộ nhớ dài hạn – Quên

- Có hai học thuyết chính về sự quên:
 - sự phân rã (decay) : thông tin trong bộ nhớ sẽ dần dần bị mất đi (Ebbinghaus ,1885)
 - sự can thiệp – nhiễu (interference): Các thông tin cũ bị mất đi do có sự can thiệp của các thông tin mới. Các thông tin cũ nhiều khi cũng có thể can thiệp lại thông tin mới

Bộ nhớ dài hạn – Quên

Sự can thiệp

- V.d.: bạn có số ĐT di động mới, việc nhớ số di động mới sẽ làm bạn quên đi số di động cũ
- V.d.: thỉnh thoảng bạn rẽ nhầm vào đường nhà người yêu cũ thay vì đi thẳng để đến nhà người yêu mới

Bộ nhớ dài hạn – Quên

- Cảm xúc cũng ảnh hưởng đến sự quên – các sự kiện có nhiều cảm xúc sẽ ít bị quên hơn các sự kiện có ít cảm xúc
- Câu hỏi đặt ra: chúng ta có thực sự quên hay chỉ là thông tin khó có thể (không thể) truy cập được? - không ai chứng minh được – tuy nhiên có những bằng chứng là thông tin có thể không mất đi hẳn

Bộ nhớ dài hạn – Truy cập thông tin

- Hai loại truy cập thông tin:
 - Nhớ lại: các thông tin được sao chép lại từ bộ nhớ
 - Nhận dạng: so sánh thông tin với các thông tin trong bộ nhớ.

Quá trình nhận dạng đơn giản hơn quá trình nhớ lại vì có thông tin làm gợi ý: v.d. nhớ ra mặt một người khó hơn nhận ra người đó; v.d. câu hỏi lựa chọn dễ hơn câu hỏi thông thường

Con người

- Mô hình hóa con người:
 - Thành phần vào/ra
 - Bộ nhớ
 - **Bộ xử lý**

Con người - Xử lý thông tin

- Suy nghĩ
- Rèn luyện kỹ năng ■
- Xử lý lỗi

Xử lý thông tin – suy nghĩ

- Điểm khác biệt giữa người và loài vật, giữa người và máy
- Suy nghĩ: giải quyết cả những vấn đề chúng ta chưa thấy bao giờ
- Suy nghĩ:
 - Suy luận (Reasoning)
 - Giải quyết vấn đề (Problem solving)

Xử lý thông tin – suy nghĩ suy luận

- Dùng các kiến thức đã có để suy ra kết luận hoặc suy ra một điều gì đó mới về lĩnh vực đang quan tâm
- Ba loại suy luận:
 - suy diễn (deductive)
 - quy nạp (inductive)
 - ??? (abductive)

Xử lý thông tin – suy nghĩ suy luận – suy diễn

- Đưa ra kết luận tất yếu từ một số giả thiết

Nếu là chiều thứ 2 thì chúng ta có tiết HCI

Hôm nay là thứ 2

=> chúng ta có tiết HCI

Xử lý thông tin – suy nghĩ suy luận – suy diễn

- Chúng ta cũng có khi suy diễn sai:

Một số học sinh cao học đi làm

Một số người đi làm là sếp

=> Một số học sinh là sếp

Xử lý thông tin – suy nghĩ suy luận – quy nạp

- Tổng quát hóa từ những trường hợp chúng ta đã thấy để suy ra những trường hợp chúng ta chưa gặp
- V.d.: Mọi người đàn ông chúng ta gặp đều sợ vợ
=> mọi người đàn ông đều sợ vợ
- Tất nhiên, suy luận này có thể sai ! Chứng minh là sai thì dễ - chỉ ra một ví dụ sai ! Không thể chứng minh đúng -> tìm càng nhiều bằng chứng càng tốt để hỗ trợ cho ý kiến

Xử lý thông tin – suy nghĩ suy luận – **abductive**

- Suy luận từ một thực tế ra hành động hay trạng thái gây ra thực tế đó

Nếu ông A uống rượu thì ông A đi xe máy nhanh
Hôm nay ông A đi xe máy nhanh

⇒ ông A uống rượu

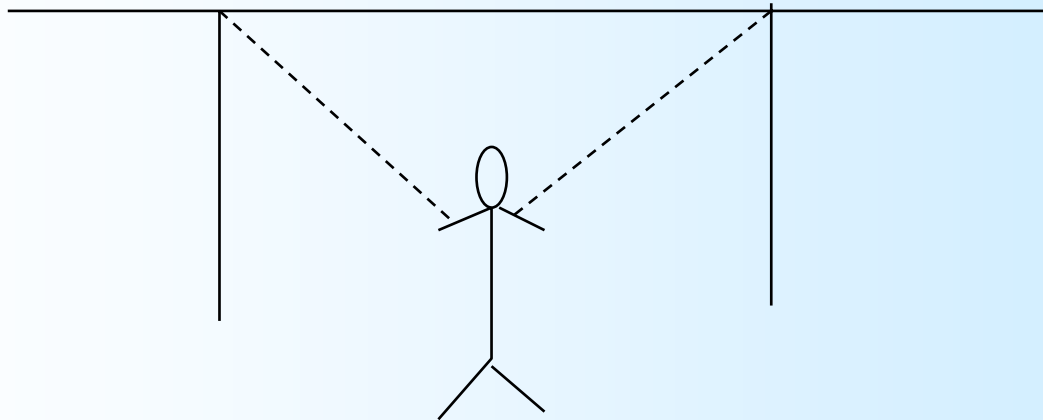
Không đáng tin cậy! Nhưng vẫn được dùng nhiều

Xử lý thông tin – suy nghĩ giải quyết vấn đề

- Tìm giải pháp cho một nhiệm vụ mới dùng những kiến thức chúng ta đã có
- Có nhiều quan điểm về việc chúng ta giải quyết vấn đề thế nào:
 - Nguyên lý Gestalt
 - Nguyên lý “không gian vấn đề”
 - Nguyên lý “phép loại suy trong giải quyết vấn đề”

Giải quyết vấn đề nguyên lý Gestalt

- Chúng ta giải quyết vấn đề bằng phương pháp mò mẫm (trial and error)
- Thí nghiệm: - cho một số cọc, một số chiếc kim – làm thế nào để nối 2 đoạn dây điện lại với nhau ?



Giải quyết vấn đề nguyên lý Gestalt

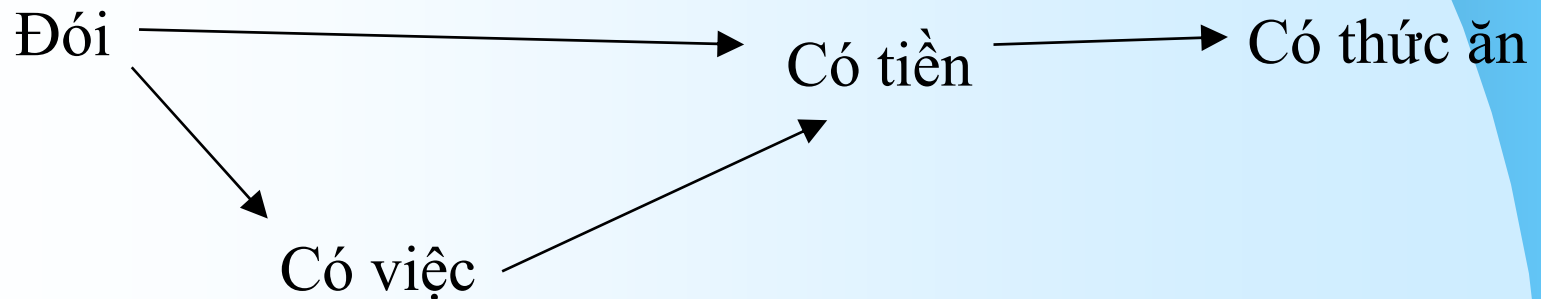
- Kết quả:
 - Có nhiều lời giải khác nhau
 - Trong đó có một lời giải: Khi người thí nghiệm đang dùng kìm để cắt vỏ dây điện – dây lắ => Buộc kìm vào dây điện - tạo ra con lắ – chạy sang phía dây kia – bắt con lắ – nối 2 dây với nhau
- Nguyên lý này nghe rất hấp dẫn – nhưng chưa có đủ bằng chứng để chứng minh

Giải quyết vấn đề nguyên lý “không gian vấn đề”

- Tâm điểm của việc giải quyết vấn đề chính là một “không gian vấn đề” (Newell và Simon, 1972).
- Không gian vấn đề chứa những trạng thái vấn đề, giải quyết vấn đề được thực hiện thông qua việc sinh ra những trạng thái vấn đề bằng các toán tử (operator) chuyển trạng thái
- Một vấn đề có một trạng thái ban đầu và một trạng thái đích. Con người dùng các toán tử để đi đến trạng thái đích.

Giải quyết vấn đề nguyên lý “không gian vấn đề”

- Tuy nhiên không gian vấn đề rất lớn nên con người dùng phương pháp tìm kiếm *heuristics*



Giải quyết vấn đề

Nguyên lý “phép loại suy trong giải quyết vấn đề”

- Phép loại suy (analogy): quá trình suy luận dựa trên sự giống nhau, sự tương tự
- Trong giải quyết vấn đề, các lĩnh vực cũ cùng các cách giải quyết cũ được đưa sang các vấn đề mới tương tự

Giải quyết vấn đề

Nguyên lý “phép loại suy trong giải quyết vấn đề”

- V.d.: Đưa cho các bác sỹ giải quyết một khối u bằng chiếu tia. Tuy nhiên nếu chiếu tia mạnh quá – phá cả các mô lành. Nếu chiếu tia yếu quá – không đủ phá khối u.
- Giải pháp: chiếu các tia yếu từ mọi phía vào khối u – vừa không phá các mô lành, vừa hợp thành tia đủ mạnh để phá khối u

Giải quyết vấn đề

Nguyên lý “phép loại suy trong giải quyết vấn đề”

- Chỉ có 10% số bác sỹ đưa ra giải pháp đó
- Tuy nhiên sau khi cho các bác sỹ xem tình huống:

Để đánh một pháo đài, người ta cần cử một sư đoàn đến. Tuy nhiên trên đường đi, có rất nhiều mìn và rất dễ nổ nếu cả sư đoàn đi qua. Người ta đành chia sư đoàn thành các nhóm nhỏ đi theo những ngả đường khác nhau.

đã có đến 80% các bác sỹ đưa ra giải pháp nói trên để xử lý khối u

Xử lý thông tin – rèn luyện kỹ năng

- Giải quyết vấn đề tập trung vào việc giải quyết những vấn đề mới
- Trên thực tế, chúng ta rất hay gặp lại các vấn đề cũ – thông qua việc lặp đi lặp lại giải quyết các vấn đề cũ -> rèn luyện kỹ năng -> tăng hiệu suất công việc
- V.d.: Nấu ăn, chơi cờ

Xử lý thông tin – Xử lý lỗi

- Trong cuộc sống, chúng ta không phải là không mắc sai lầm ! Có sai lầm nhỏ, có sai lầm lớn
- Lỗi khi xử dụng kỹ năng: khi tình huống đã thay đổi, chúng ta vẫn sử dụng kỹ năng đó => gây ra lỗi. V.d.: Kỹ năng đi xe từ chỗ làm về nhà. Hôm nay vợ dặn qua siêu thị mua đồ - vẫn dùng kỹ năng cũ -> đi quá mất siêu thị

Xử lý thông tin – Xử lý lỗi

- Lỗi khi hiểu sai về một vấn đề, một hệ thống.

Tại sao? Con người thường xây dựng một mô hình trong bộ nhớ về các hệ thống. Các mô hình này thường không đầy đủ, thậm chí không nhất quán vì không được xem xét kỹ lưỡng => gây ra lỗi

Tâm lý học và HCI

- Làm sao chúng ta có thể ứng dụng những thứ vừa học vào việc thiết kế các hệ thống tương tác?
-
- Lưu ý: những thứ chúng ta vừa đề cập đến là nói về con người nói chung. Tuy nhiên chúng ta phải để ý rằng mọi người đều khác nhau => thiết kế các hệ thống phù hợp với từng người khác nhau

Tâm lý học và HCI

- V.d.: Màu xanh nước biển không nên dùng để thông báo các thông tin quan trọng – nên dùng màu đỏ
- Tuy nhiên, không nên áp dụng những thông tin này trực tiếp vào việc thiết kế - vì chúng không đầy đủ và đơn giản
- Thay vào đó, chúng ta sẽ áp dụng những nguyên tắc và hướng dẫn thiết kế được suy ra từ các học thuyết tâm lý học

Tâm lý học và HCI

- Ngoài ra, các học thuyết tâm lý học còn giúp chúng ta xây dựng những mô hình để mô phỏng người sử dụng => thiết kế tốt hơn
- Cuối cùng, các học thuyết tâm lý học giúp chúng ta đánh giá thiết kế và hệ thống của chúng ta

Tâm lý học và HCI

Bộ nhớ

- Sử dụng kiến thức có sẵn khi có thể
- Giúp người dùng sử dụng nhận dạng thay vì nhớ lại:
 - Dùng menu thay vì giao diện lệnh
- Không nên làm quá tải bộ nhớ ngắn hạn (bộ nhớ làm việc) của người dùng:
 - v.d. không bắt người dùng nhớ nhiều quá – lưu trữ trên bộ nhớ máy tính – current settings
- Thiết kế các giao tiếp một cách nhất quán:
 - tránh sự can thiệp (nhiều) giữa thông tin cũ và mới

Tâm lý học và HCI

Khả năng nhận thức

- Tận dụng mọi khả năng của con người
 - v.d. sử dụng nhiều màu thay vì chỉ hai màu
- Sử dụng nhiều phương thức truyền khác nhau:
 - v.d. cả âm thanh và hình ảnh
- Làm nhẹ bớt những nhiệm vụ khó
 - v.d. pop-up menu để giúp cho mắt đỡ phải chuyển động

9

CÁC KỸ THUẬT ĐÁNH GIÁ

Nội dung

- 9.1. Khái niệm và vai trò của đánh giá trong thiết kế HCI
- 9.2. Các kiểu đánh giá
 - 9.2.1. Đánh giá trong phòng thí nghiệm
 - 9.2.2. Đánh giá tại chỗ
- 9.3. Đánh giá thiết kế (mang tính dự đoán)
 - 9.3.1. Mô hình Cognitive Walkthrough
 - 9.3.2. Đánh giá dựa vào mô phỏng
 - 9.3.3. Đánh giá dựa vào xem xét lại quá trình (Review based)
 - 9.3.4. Đánh giá dựa vào mô hình
- 9.4. Đánh giá cài đặt
 - 9.4.1. Kỹ thuật thực nghiệm
 - 9.4.2. Kỹ thuật quan sát
 - 9.4.3. Kỹ thuật hỏi đáp
- 9.5. Lựa chọn phương pháp
 - 9.5.1. Tiêu chí lựa chọn phương pháp
 - 9.5.2. Quy trình 4 bước

9.1. Khái niệm và vai trò của đánh giá trong thiết kế HCI

Đánh giá không phải là một giai đoạn trong quá trình thiết kế mà là nhiệm vụ trung tâm của vòng đời thiết kế và diễn ra trong suốt vòng đời của HCI (hình 0.3 phần 0.4 - chương 0). Tuy nhiên không thể thực hiện các kiểm thử thực nghiệm trong suốt quá trình thiết kế mà chỉ có thể hoặc nên dùng các kỹ thuật phi hình thức hay phân tích. Như vậy sẽ có mối liên hệ hẹp giữa đánh giá, mô hình hoá và mẫu thử, các kỹ thuật hỗ trợ đảm bảo rằng thiết kế được đánh giá liên tục.

Đánh giá là thu thập dữ liệu kiểm tra về tính dùng được của thiết kế và đảm bảo ba nhiệm vụ chính:

- i) Khẳng định tính mở rộng các chức năng của hệ thống
- ii) Khẳng định tính hiệu quả của giao tiếp đối với người dùng.
- iii) Xác định một số vấn đề đặc biệt nảy sinh trong quá trình sử dụng.

Ở đây, chức năng của hệ thống là quan trọng và nó phải đáp ứng với đặc tả yêu cầu người dùng. Thiết kế hệ thống phải có khả năng đáp ứng các nhiệm vụ đặt ra một cách dễ dàng. Điều này có nghĩa thiết kế không chỉ tạo nên sự hiện diện các chức năng tương ứng mà còn có thể với được bởi người dùng theo nghĩa đó là hành động mà người dùng cần để thực hiện nhiệm vụ. Nó cũng bao hàm việc đánh giá khả năng sử dụng của hệ thống với những cái mà người dùng mong đợi.

Tóm lại, ngoài việc đánh giá hệ thống về chức năng, việc đánh giá cũng cần có khả năng đo đếm sự ảnh hưởng của hệ thống đối với người dùng. Nó bao hàm nhiều khía cạnh như hệ thống có dễ dùng, dễ học,...? Ngoài ra, các vấn đề khác như tải của hệ thống đối với người dùng, người dùng có cần phải nhớ một lượng thông tin không lồ không?

9.2. Các kiểu đánh giá

Hiện tại có nhiều kỹ thuật dùng để đánh giá một hệ thống tương tác. Tiêu chí phân loại cũng khá đa dạng. Nếu dựa vào điều kiện môi trường, nơi tiến hành đánh giá, người ta phân các kỹ thuật đánh giá ra thành hai nhóm chính: đánh giá trong phòng thí nghiệm và đánh giá thực địa. Nếu phân theo thời gian, vòng đời của quá trình thiết kế ta cũng có hai nhóm: đánh giá thiết kế và đánh giá cài đặt. Mỗi phương pháp có những đặc điểm riêng, cách áp dụng riêng tuy đều nhằm các mục đích nêu trên.

9.2.1. Đánh giá trong phòng thí nghiệm

Việc đánh giá này diễn ra trong phòng thí nghiệm và thường dùng trong quá trình thiết kế. Trong cách đánh giá này, người đánh giá muốn thực hiện một số khẳng định của mình và không cần sự có mặt của người dùng. Tuy nhiên, người dùng cũng có thể có mặt tại đó và tham gia vào quá trình đánh giá.

Do tiến hành trong phòng thí nghiệm nên có điều kiện khách quan hơn. Nhất là các phòng thí nghiệm được trang bị tốt để nghiên cứu tính dùng được như các thiết bị nghe nhìn, camera, gương phản chiếu,..., cái sẽ không có trên thực địa. Tuy nhiên, nhược điểm lớn nhất của phương pháp này là thiếu ngữ cảnh, môi trường không tự nhiên, không có thật. Và đặc biệt khó khăn khi phải quan sát nhiều người cùng cộng tác trên một nhiệm vụ trong tình huống phòng thí nghiệm khi mà việc giao tiếp giữa con người với nhau lại rất phụ thuộc

Chương 9: Các kỹ thuật đánh giá giao tiếp người dùng

vào ngữ cảnh. Việc đánh giá này đặc biệt là cần thiết khi môi trường thực địa không cho phép, thí dụ như khi điều kiện môi trường nguy hiểm hay ở xa như các trạm vũ trụ. Hơn nữa, nếu chúng ta mong muốn phát hiện một số vấn đề hay một số thủ tục ít dùng hay so sánh các thiết kế khác nhau thì đây là cách tốt nhất.

9.2.2. Đánh giá tại chỗ

Việc đánh giá được tiến hành với sự tham gia của người dùng. Nó có thể diễn ra trong giai đoạn thiết kế hay cài đặt. Việc đánh giá thực hiện trong môi trường người dùng nhằm đánh giá hệ thống trong hoạt động và trạng thái người dùng. Tuy nhiên, do tiến hành trên thực địa sẽ có nhiều yếu tố gây ảnh hưởng như tiếng ồn, người chuyển động đi lại, ... sẽ gây mất tập trung, do vậy đánh giá có thể có thiếu sót. Bù lại, do bản chất tự nhiên, khách quan, chúng ta sẽ quan sát được sự tương tác giữa hệ thống, giữa các cá nhân, điều không thể có khi tiến hành trong phòng thí nghiệm. Ngữ cảnh là ngữ cảnh thật và người dùng cũng làm việc trong môi trường thật.

Đánh giá trên thực địa tỏ ra ưu việt hơn khi ta có thể nghiên cứu sự tương tác cũng như những gì xảy ra trong ngữ cảnh thật. Ngay cả việc tạm ngừng cũng quan trọng khi nó sẽ phơi bày các trạng thái đặc biệt như khi lưu trữ hay khôi phục trong một nhiệm vụ. Tất nhiên, cũng cần chú ý là do sự hiện diện của người đánh giá mà người dùng sẽ mất tập trung, thao tác có thể thiếu chính xác.

9.3. Đánh giá thiết kế

Như đã đề cập ở trên, việc đánh giá có thể diễn ra ngay trong quá trình thiết kế. Những đánh giá đầu tiên về hệ thống, một cách lý tưởng nên thực hiện trước khi hệ thống được cài đặt. Nếu trong quá trình thiết kế này, nhờ đánh giá lỗi được phát hiện sớm sẽ có lợi tránh cho những ảnh hưởng đáng tiếc. Điển hình là nếu lỗi được phát hiện muộn hơn trong quá trình thiết kế thì chi phí cho chỉnh sửa sẽ cao hơn.

Người ta thường sử dụng một số kỹ thuật sau đây để đánh giá trong quá trình thiết kế:

- Cognitive WalkThrough
- Đánh giá kiểu Heuristic
- Đánh giá dựa vào mô hình
- Đánh giá dựa vào xem xét lại

Phần lớn các đánh giá này không cần sự có mặt của người dùng. Tuy vậy, nó phụ thuộc vào người đánh giá, hay các chuyên gia, người làm nhiệm vụ thiết kế hay khẳng định tác động sẽ xảy ra với một người dùng đặc biệt nào đấy. Khuynh hướng chủ yếu của đánh giá trong giai đoạn này nhằm xác định một lĩnh vực nào đó có thể là nguyên nhân gây khó khăn do vi phạm nguyên tắc của nhận thức hay bỏ qua kết quả thực nghiệm chấp nhận được.

9.3.1. Mô hình Cognitive Walkthrough

Cognitive Walkthrough do Polson và các đồng nghiệp của ông đề xuất khi thử giới thiệu lý thuyết tâm lý học vào kỹ thuật walkthrough khách quan, phi hình thức. Nó là

phương pháp có tính dự đoán kiểu Review nhằm phát hiện vấn đề từ rất sớm. Walkthrough yêu cầu xem xét một cách chi tiết chuỗi các hành động, mỗi hành động tương tự như một đoạn mã của chương trình. Người đánh giá theo các bước định nghĩa từ các nhiệm vụ đặc tả hệ thống hoặc từ màn hình "mock-up": từ màn hình này qua màn hình khác nhằm kiểm tra tính đúng được. Thường, điểm nhấn chính của kỹ thuật này là thiết lập tình để học và để dùng của hệ thống thông qua cách học theo kiểu "khám phá". Kinh nghiệm chỉ ra rằng có nhiều người thích học theo kiểu này hơn là qua đào tạo hay tra trong tài liệu hướng dẫn. Do vậy, cách kiểm tra tiến hành trong quá trình walkthrough là đưa ra các câu hỏi liên quan đến việc khám phá. Để đánh giá, kỹ thuật này cần các thông tin sau:

- i) Một mô tả về nguyên mẫu của hệ thống. Mô tả này không cần đầy đủ song cũng nên khá chi tiết, thí dụ như vị trí và ngôn từ cho menu.
- ii) Một mô tả về nhiệm vụ mà người dùng phải thực hiện. Nhiệm vụ phải mang tính tiêu biểu, cái mà người dùng hay làm.
- iii) Một danh sách chi tiết các hành động cần thiết để hoàn thành nhiệm vụ theo nguyên mẫu.
- iv) Một chỉ dẫn về người dùng là ai và các tri thức, kinh nghiệm mà người đánh giá có thể giả định.

Với thông tin trên, người đánh giá thực hiện chuỗi các hành động và đưa ra các ý kiến về tính đúng được. Với mỗi hành động, người đánh giá cố gắng trả lời bốn câu hỏi:

- i) Người dùng có tạo ra bất cứ tác động gì trên hành động đó không? Những khẳng định về nhiệm vụ mà hành động được hỗ trợ đúng theo kinh nghiệm và tri thức của người dùng trên tương tác đó.
- ii) Người dùng có khả năng để ý hành động đúng là có?. Thí dụ, người dùng có nhìn thấy một phím hay một mục của menu, qua đó hành động tiếp là đang thực hiện bởi hệ thống. Điều này có nghĩa là người dùng sẽ thấy phím đó (hay mục đó) tại thời điểm họ cần (không nhất thiết phải biết).
- iii) Khi người dùng tìm thấy một hành động đúng trên giao tiếp liệu họ có biết đó là cái duy nhất đúng cho mục đích mà họ cố tạo ra? Có thể đó là một phím lệnh/ mục menu quan sát được nhưng liệu người dùng có biết đó là cái mà họ cần để hoàn thành nhiệm vụ.
- iv) Sau khi hành động tiến hành, người dùng sẽ hiểu phản hồi của hệ thống? Giả sử rằng người dùng đã chọn đúng hành động, họ sẽ biết tiếp cái gì? Đó là khái niệm về chu trình tương tác *đánh giá / thực hiện*.

Để tiến hành đánh giá, người đánh giá phải chuẩn bị và ghi theo mẫu biểu sau:

Bảng thực hiện đánh giá nhận thức Walkthrough

Giao diện:

Nhiệm vụ:

Chương 9: Các kỹ thuật đánh giá giao tiếp người dùng

Người đánh giá: Ngày tháng: . . . / . . . / . . .

Mô tả nhiệm vụ:

Chuỗi hành động:

Đoán trước của người dùng :

Đích ban đầu của người dùng :

Hành động tiếp theo: Description

1. Hiệu chỉnh mục tiêu (đích)
2. Các vấn đề cho hiệu chỉnh mục tiêu

-

-

3. Vấn đề xác định hành động

-

-

4. Vấn đề thực hiện hành động

-

-

Đề minh họa việc đánh giá theo kỹ thuật này, chúng ta xét thí dụ sau: **Thiết kế cái điều khiển từ xa cho đầu Video.**

Mục tiêu: Thiết kế điều khiển từ xa cho Video nhằm cho phép người dùng lập trình chọn lựa thời gian ghi.

Thiết kế ban đầu: hình 9.1a, b

- Hình 9.1a: Tập các phím trong trạng thái sử dụng bình thường.
- Hình 9.1b: Phím ghi thời gian được nhấn.

Video cho phép người dùng lập trình ba chế độ ghi theo chuỗi khác nhau. Số hiệu chuỗi tiếp theo có thể được tự động thiết lập. Chúng ta muốn biết liệu thiết kế của mình có hỗ trợ nhiệm vụ người dùng không?

Chúng ta bắt đầu bằng việc xác định nhiệm vụ tiêu biểu. Giả sử người dùng quen thuộc với ghi hình nhưng không biết thiết kế chuyên dụng này. Bước tiếp theo của Walkthrough là xác định chuỗi hành động cần phải tiến hành. Chúng ta đặc tả điều này theo thuật ngữ hành động người dùng (Action) và đáp ứng của hệ thống (System Response):

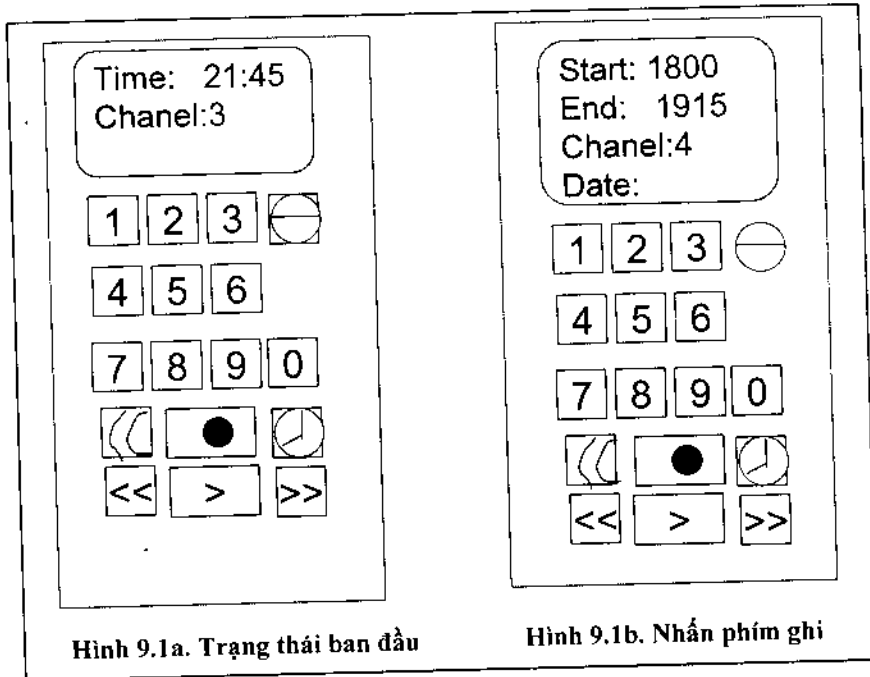
Action A: Nhấn phím thời gian ghi

Đáp ứng A: Chuyển về hiển thị theo mode thời gian. Con trỏ xuất hiện sau chữ Start.

Chương 9: Các kỹ thuật đánh giá giao tiếp người dùng

Action B: Nhấn số 1800

Đáp ứng B: Mỗi chữ số được hiện ngay sau khi nhấn và con trỏ chuyển về vị trí tiếp.



Action C: Nhấn phím ghi thời gian.

Đáp ứng C: Con trỏ chuyển về sau vị trí End.

Action D: Nhấn số 1915.

Đáp ứng D: Mỗi chữ số được hiện ngay sau khi nhấn và con trỏ chuyển về vị trí tiếp.

Action E: Nhấn phím ghi thời gian.

Đáp ứng E: Con trỏ chuyển về sau vị trí Chanel.

Action F: Nhấn số 4.

Đáp ứng F: Mỗi chữ số được hiện ngay sau khi nhấn và con trỏ chuyển về vị trí tiếp.

Action G: Nhấn phím ghi thời gian.

Đáp ứng G: Con trỏ chuyển về sau vị trí Date.

Action H: Nhấn số 010199.

Đáp ứng H: Mỗi chữ số được hiện ngay sau khi nhấn và con trỏ chuyển về vị trí tiếp.

Action I: Nhấn phím ghi thời gian.

Đáp ứng I: Số hiệu chuỗi hiện lên góc phải trên màn hình.

Action J: Nhấn phím truyền.

Đáp ứng J: Dữ liệu được truyền đi và chuyển về chế độ bình thường.

Chương 9: Các kỹ thuật đánh giá giao tiếp người dùng

Sau khi đã thiết lập chuỗi hành động, tiến hành đánh giá bởi Walkthrough. Với mỗi hành động A, B,..., J, ta phải trả lời bốn câu hỏi sau và nói về tính dùng được của hệ thống. Chúng ta bắt đầu với hành động A:

Action A: Nhấn phím ghi thời gian

Câu hỏi 1: Người dùng có cố gắng tạo nên bất cứ tác động nào khi hành động xảy ra? Giao tiếp không cung cấp chỉ dẫn nào khi người dùng nhấn phím ghi thời gian. Tuy nhiên, điều đó chấp nhận được vì giả thiết là người dùng quen thuộc với máy ghi Video

Câu hỏi 2: Người dùng có khả năng để ý hành động đúng là có? Phím ghi thời gian có trên điều khiển từ xa.

Câu hỏi 3: Khi người dùng tìm thấy một hành động đúng trên giao tiếp liệu họ có biết đó là cái duy nhất đúng cho mục đích mà họ cố tạo ra?

Điều này không rõ ràng với phím ghi thời gian vì biểu tượng đồng hồ trên điều khiển từ xa có thể là ứng cử viên, tuy nhiên nó có thể hiểu là phím dùng để thay đổi thời gian. Ứng cử viên khác có thể như là phím thứ 4 bên trái (phím này liên quan tới ghi). Tóm lại, biểu tượng đồng hồ là lựa chọn đúng, song rất có thể người dùng bị nhầm tại điểm này.

Câu hỏi 4: Sau khi hành động tiến hành, người dùng sẽ hiểu phản hồi của hệ thống?

Ngay sau khi hành động được làm, việc hiển thị đã thay đổi chế độ và các dòng hiện lên là khá thân thuộc với người dùng (Start, End,...). Cứ theo cách như vậy tiến hành đánh giá cho các hành động khác.

9.3.2. Đánh giá Heuristic

Là các nguyên tắc chung, hay các guideline hay các luật có thể trợ giúp một quyết định trong thiết kế, hoặc dùng để phê phán một quyết định đã được làm. Kỹ thuật này do Jacob Nielsen và Rolf Molich đề xuất và được sử dụng trong giai đoạn đầu của thiết kế nhằm cố định tính dùng được. Ý tưởng chính ở đây là nhiều người đánh giá độc lập cùng tiến hành trên một hệ thống để nêu lên tính dùng được. Kỹ thuật này tốt nhất nên sử dụng trong giai đoạn đầu của thiết kế vì nó cố định nhiều vấn đề của tính dùng được. Tuy nhiên, mọi cái yêu cầu được đánh giá cần được mô tả theo kiểu storyboard để cung cấp một cái nhìn tổng quan về hệ thống hoặc có thể đầy đủ nếu được. Để hỗ trợ quá trình đánh giá, Nielson đưa ra 9 heuristics và 10 quy định.

9 heuristics

- 1) Đối thoại đơn giản và tự nhiên
 - Đơn giản: sử dụng ít thông tin.
 - Tự nhiên: lệnh gần với yêu cầu.
- 2) Nói ngôn ngữ của người dùng
 - Sử dụng cách nói của người dùng.
 - Không dùng thuật ngữ công nghệ đặc biệt.

Chương 9: Các kỹ thuật đánh giá giao tiếp người dùng

3) Giảm tải tối thiểu cho bộ nhớ người dùng

- Không bắt người dùng phải nhớ cho hành động tiếp sau.
- Lưu lại thông tin trên màn hình cho đến khi người dùng không cần nữa.

4) Tính nhất quán

- Dãy các hành động phải học trong phần này phải áp dụng được cho các phần khác.

5) Phản hồi thông tin

- Người dùng nhận biết được tác động của hành động mình đã thực hiện trên hệ thống.

6) Xử lý tình huống thoát

- Khi người dùng gặp một tình huống/ vấn đề mà họ không quan tâm, họ có thể thoát ra một cách nhanh chóng và không làm tổn hại đến bất cứ một cái nào khác.

7) Tạo lối tắt

- Giúp đỡ người dùng có kinh nghiệm tránh những đối thoại và thông tin không cần thiết.

8) Có thông báo lỗi tốt

- Giúp người dùng hiểu nguyên nhân của sự việc đó và có cách để khắc phục.

9) Dự báo lỗi

10 quy định

- 1) Thấy được các trạng thái của hệ thống : hệ thống luôn cho người dùng thấy cái họ sẽ làm qua thông tin phản hồi trong thời gian hợp lý.
- 2) Sự tự do và quản lý người dùng: người dùng có thể gặp lỗi, do vậy phải có thoát khẩn cấp.
- 3) Dự báo lỗi
- 4) Chuẩn hoá và nhất quán
- 5) Đối sánh hệ thống với thế giới thực
- 6) Nhận biết chứ không phải nhớ lại: tạo cho đối tượng, hành động và các lựa chọn dễ dàng nhận biết bởi người dùng chứ không cần nhớ.
- 7) Sử dụng hiệu quả và mềm dẻo: Dùng được cho cả hai loại người dùng có và không có kinh nghiệm.
- 8) Thiết kế đơn giản và có thẩm mỹ
- 9) Trợ giúp người dùng nhận biết, đối thoại và phục hồi từ trạng thái lỗi.
- 10) Trợ giúp và tài liệu.

Chương 9: Các kỹ thuật đánh giá giao tiếp người dùng

9.3.3. Đánh giá dựa vào xem xét lại quá trình (Review based)

Nó cũng là đánh giá kiểu Heuristic do Molich & Nelson đưa ra 1990: dùng trong các ứng dụng nhỏ, thời gian đánh giá chừng 2 giờ. Tuy nhiên có thể nhiều hơn nếu ứng dụng lớn. Nó cũng là đánh giá kiểu Discount usability: chi phí thấp, thời gian và tài nguyên không nhiều.

9.3.4. Đánh giá dựa vào mô hình

Một số các mô hình nhận thức và thiết kế cung cấp một phương tiện để tổ hợp cho đặc tả thiết kế và đánh giá trong cùng một khung tương tác (chương 5). Nó là kỹ thuật đánh giá đòi hỏi:

- Đặc tả chức năng của hệ thống phân có liên quan.
- Một phân tích nhiệm vụ chứa danh sách nhiệm vụ và gán chúng thành các thành phần.
- Cấu trúc nhiệm vụ từ đơn giản đến phức tạp.
- Các thao tác người dùng có thể đánh giá bằng phương pháp giải tích.

Trong kiểu đánh giá này, mô hình vật lý Keyrock hay được áp dụng. Chúng ta sẽ minh họa thông qua thí dụ "Save File trong MS Word".

Save File trong MS Word

- Phương tiện: Dùng chuột và menu pull down

- Giá thiết: Nhiệm vụ bắt đầu bằng thao tác Homming khi người dùng đặt tay lên chuột. Thời gian nhấn phím là 0,35 s, thời gian đáp ứng của hệ thống là 1,2 s.

- Dãy thao tác:

1) Bắt đầu T_H (người dùng đặt tay lên chuột và điều khiển được chuột).

2) Chuyển con trỏ đến menu trên đỉnh màn hình: T_P , cộng với thời gian suy nghĩ T_M .

Tổng thời gian thực hiện: $T_P + T_M$

3) Chọn trên menu File (nhấn File), dịch chuyển con trỏ rồi chọn Save:

Tổng thời gian thực hiện: $T_M + T_K(\text{File}) + T_P + T_K(\text{Save})$

4) MS Word nhắc nhập tên tệp: T_R ; người dùng nhấn "File-2.4", cuối cùng nhấn phím Enter: $T_K(\text{return})$:

Tổng thời gian thực hiện: $T_R + T_M + T_K(\text{File-2.4}) + T_K(\text{return})$

Với $T_H = 0,4$, $T_M = 1,35$, $T_P = 1,1$, $T_K = 0,35$, $T_R = 1,2$, và như vậy, tổng thời gian thực hiện là 13,05 s.

9.4. Đánh giá cài đặt

Đây là kỹ thuật đánh giá có sự hiện diện của người dùng và hệ thống đã được cài đặt. Người ta có thể sử dụng mô phỏng hay mẫu thử. Ba kỹ thuật chính tiêu biểu sau đây hay được dùng:

- i) Đánh giá thực nghiệm.
- ii) Kỹ thuật quan sát.

Chương 9: Các kỹ thuật đánh giá giao tiếp người dùng

iii) Kỹ thuật hỏi đáp.

9.4.1. Đánh giá thực nghiệm

Là một trong các phương pháp quan trọng để đánh giá thiết kế hay khía cạnh nào đó của thiết kế. Nó cung cấp một chứng cứ thực nghiệm để hỗ trợ một yêu cầu hoặc một giả thiết cụ thể. Mọi phương pháp thực nghiệm đều có cùng một dạng cơ bản.

Cách thức tiến hành: Người đánh giá lựa chọn giả thiết đánh giá thông qua một số thuộc tính của chủ thể như các biến độc lập hay phụ thuộc và nhằm đánh giá một số khía cạnh đặc trưng của thiết kế. Việc điều khiển đánh giá được thay đổi thông qua giá trị một số biến. Ba thông số quan trọng trong kỹ thuật này là : chủ đề (subjects), biến (variables) và giả thiết (hypothesis).

a) Chủ đề

Việc lựa chọn chủ đề quyết định tới thành công của đánh giá. Chủ đề chọn phải sánh được với kỳ vọng của lớp người dùng, càng gần càng tốt. Lý tưởng là kiểm tra với sự hiện diện của người dùng. Tuy nhiên không phải luôn có thể thực hiện được. Nếu người dùng không phải là người dùng thực sự thì phải chọn lớp tương tự: độ tuổi, tri thức, ...

Vấn đề thứ hai liên quan đến tập chủ đề đó là kích thước của tập mẫu. Kích thước này thường được xem xét bởi tính thực thi như sự hiện diện của chủ thể thường bị giới hạn hay tính khan hiếm của tài nguyên,... Tuy nhiên, mẫu chọn thường phải đủ lớn và đặc trưng (ít nhất là 10).

b) Biến/đại lượng

Thực nghiệm quản lý và đo các đại lượng dưới điều kiện được kiểm soát nhằm kiểm tra giả thiết. Người ta hay sử dụng hai loại biến (đại lượng): loại quản lý và loại đo. Loại thứ nhất gọi là biến độc lập, loại thứ hai là phụ thuộc.

Biến độc lập là các đặc trưng của thực nghiệm, nhằm tạo ra các điều kiện so sánh khác nhau như: kiểu giao tiếp, mức độ trợ giúp, số mức menu, các icon, ... Với các kiểm tra phức tạp thì phải cần nhiều biến độc lập. Mỗi biến có thể cung cấp các giá trị khác nhau và trong đánh giá thực nghiệm chúng được biết đến với tên "mức độ của biến".

Biến phụ thuộc là các biến có thể đo đếm được theo nhiều cách. Thí dụ: tốc độ lựa chọn menu trong giao tiếp dùng menu. Thường trong kỹ thuật đánh giá thực nghiệm, các biến phụ thuộc thường được chọn để biểu diễn các độ đo như thời gian thực hiện nhiệm vụ, số lỗi phạm phải,... Chú ý là càng nhiều số đo chủ quan được áp dụng để đối sánh với các số đo định trước càng tốt và là yếu tố quan trọng phải xem xét.

c) Giả thiết

Giả thiết là một số dự đoán kết quả của thực nghiệm được hình thành từ biến độc lập và biến phụ thuộc: thí dụ như thay đổi giá trị biến độc lập sẽ tạo nên sự thay đổi trong biến độc lập. Mục đích của thực nghiệm là chỉ ra rằng các dự đoán là chính xác. Người ta thường dùng các số đo thống kê để so sánh với các các mức độ có nghĩa khác nhau.

d) Thiết kế thực nghiệm

Để tạo ra kết quả tin cậy được và có tính khái quát, một đánh giá thực nghiệm phải được thiết kế cẩn thận. Như đã đề cập ở trên, có rất nhiều yếu tố phải xem xét, chủ yếu là chủ đề, các biến độc lập và phụ thuộc, các giả thiết. Để thiết kế đánh giá thực nghiệm, ta tiến hành qua hai bước:

1- Xem xét các yếu tố mà thực nghiệm phải coi như chủ đề, biến và giả thiết để quyết định chính xác cái bạn đang thử chứng minh.

2- Lựa chọn phương pháp sử dụng. Ở đây, có hai nhóm phương pháp chính: giữa các nhóm và trong nội bộ nhóm.

Trong phương pháp giữa các nhóm (random), mỗi một chủ đề được gán cho một điều kiện khác nhau. Phải có ít nhất hai điều kiện: điều kiện thực nghiệm (trong đó các biến được điều khiển) và điều kiện kiểm tra. Việc kiểm tra này dùng để khẳng định điều khiển sẽ chịu trách nhiệm về những cái khác nhau sẽ được đo đếm. Số lượng nhóm phụ thuộc vào số lượng biến độc lập, tuy nhiên thường chọn lớn hơn hai nhóm.

Ưu điểm của thiết kế này là mọi tác động của việc học thu được từ kết quả thực hiện của người dùng trong một điều kiện sẽ được kiểm tra. Mỗi người dùng chỉ thực hiện trong một điều kiện. Nhược điểm chính là đòi hỏi nhiều chủ đề, do vậy sự khác nhau giữa các nhóm có thể phủ nhận bất cứ kết quả nào. Vì thế, cần chọn lựa cẩn thận các chủ đề để đảm bảo đại diện cho tất cả người dùng.

Với phương pháp trong nhóm, mỗi người dùng sẽ thực hiện trong điều kiện khác nhau. Thiết kế này có thể phải chịu đựng do việc truyền tác động học. Tuy nhiên, có thể hạn chế được nếu thử tự trong đó điều kiện được người dùng xem xét bị thay đổi. Ưu điểm ở đây là chi phí thấp do cần ít người dùng và sẽ hiệu quả nếu có đào tạo.

Việc lựa chọn phương pháp phụ thuộc vào tài nguyên, chủ đề tiêu biểu,... Các kiểm tra khác nhau sẽ tạo nên các khẳng định khác nhau về dữ liệu và nếu như việc lựa chọn không đúng, thì kết quả có thể không hợp lý.

Khi chúng ta đã xác định xong giả thiết mà chúng ta sẽ cố gắng đánh giá, khi các biến đã được nghiên cứu, các chủ đề đã bố trí xong, và thiết kế là phù hợp, việc còn lại là quyết định sẽ phân tích kết quả thu được như thế nào? Phương pháp phân tích thống kê có vẻ thích hợp và có nhiều phương pháp có thể. Việc lựa chọn phương pháp phân tích nào là điều sống còn với sự thành công của đánh giá thực nghiệm. Các phân tích khác nhau tạo nên các khẳng định khác nhau về dữ liệu và nếu sự lựa chọn không phù hợp sẽ dẫn đến kết quả không hợp thức. Chúng ta sẽ xem xét một số các số đo thống kê tiêu biểu trong phần tiếp theo.

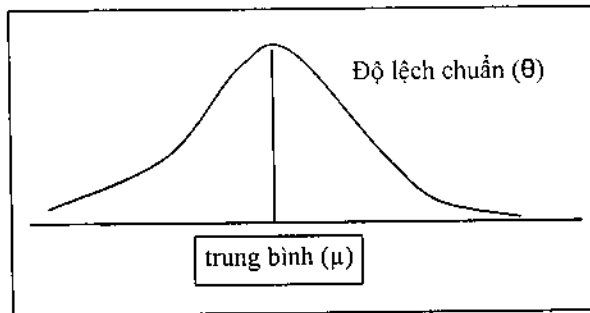
Số đo thống kê

Hai luật cho phân tích thống kê là Quan sát và Lưu trữ dữ liệu. Quan sát dễ dàng thực hiện kiểm tra thống kê qua đồ thị, lưu đồ hoặc bảng giá trị. Lưu trữ rất quan trọng cho các phương pháp phân tích về sau.

Chương 9: Các kỹ thuật đánh giá giao tiếp người dùng

Việc lựa chọn phương pháp phân tích thống kê phụ thuộc vào kiểu dữ liệu và câu hỏi mà chúng ta muốn trả lời. Các biến có thể là loại rời rạc hay liên tục. Biến rời rạc chỉ có thể nhận một tập hữu hạn các giá trị hay mức độ, thí dụ như màu màn hình: đỏ, xanh, trắng,... Biến liên tục có thể nhận vô số giá trị, thí dụ như thời gian hoàn thành nhiệm vụ. Biến liên tục có thể chuyển về rời rạc bằng cách phân lớp (thấp, trung bình, cao,...)

Biến phụ thuộc là biến đo được và tùy thuộc vào các biến thể của thực nghiệm ngẫu nhiên. Trong trường hợp biến liên tục và dữ liệu tuân theo một phân bố đã biết, một số kiểm tra thống kê đặc biệt có thể sử dụng. Các kiểm tra như thế gọi là kiểm tra có tham số và phân bố hay được dùng là phân bố chuẩn. Điều này có nghĩa là khi ta vẽ biểu đồ của sai số ngẫu nhiên, ta sẽ thu được dạng hình chuông nổi tiếng (hình 9.2).



Hình 9.2. Lược đồ sai số của phân bố chuẩn.

Trong khuôn khổ cuốn sách này, chúng tôi không trình bày chi tiết. Bạn đọc quan tâm xin mời đọc trong tài liệu tham khảo [2].

9.4.2. Kỹ thuật quan sát

Một cách phổ biến để thu thập thông tin về việc sử dụng hiện thời của hệ thống là quan sát cách người dùng đang tương tác với hệ thống. Thường người dùng được hỏi về việc thực hiện một tập các nhiệm vụ định trước và việc quan sát được tiến hành với sự hiện diện của người dùng. Nó là kỹ thuật không hình thức, có thể quan sát hay dùng video. Nếu chỉ tiến hành quan sát đơn giản thì không đủ trả lời cho sự phù hợp của hệ thống với yêu cầu của người dùng. Hai kỹ thuật hay được dùng là quan sát trực tiếp và quan sát gián tiếp.

- Quan sát trực tiếp: quan sát người dùng ở trạng thái đang làm việc, đo vậy hiệu quả và giảm được trung gian. Nhược điểm của kỹ thuật này là người dùng có thể mất tập trung vì có sự hiện diện của người đánh giá.

- Quan sát gián tiếp: quay video và sau đó phân tích kết quả thu nhận được. Kỹ thuật hay sử dụng kỹ thuật "think aloud" và đánh giá tập thể (cooperative evaluation).

Think aloud

Kỹ thuật nhằm mô tả cái người dùng tin sẽ xảy ra, tại sao lại hành động như vậy

Chương 9: Các kỹ thuật đánh giá giao tiếp người dùng

và họ sẽ cố làm cái gì? Ưu điểm của kỹ thuật này là đơn giản, đòi hỏi ít tri thức để thực hiện và cung cấp sự hiểu biết sâu sắc về các vấn đề với một giao tiếp. Nó cũng được dùng để hiểu hệ thống đang được sử dụng thế nào. Kỹ thuật này cũng có thể dùng để đánh giá xuyên suốt quá trình thiết kế bằng bút, giấy hay các màn hình mỗi trong giai đoạn đầu của thiết kế. Quá trình quan sát có thể gây tổn hại đến cách thức mà người dùng thực hiện nhiệm vụ, do vậy có thể có cái nhìn thiên lệch.

Đánh giá tập thể

Là một biến thể của think aloud, trong đó người dùng cố gắng xem mình như một thành viên trong quá trình đánh giá chứ không đơn thuần là chủ thể bị thực nghiệm. Khi bắt đầu tiến hành, người đánh giá có thể hỏi người dùng các câu hỏi như: tại sao, cái gì nếu,.... Kỹ thuật này thể hiện một số ưu điểm:

- + Quy trình ít bị ràng buộc và dễ học bởi người đánh giá.
- + Người dùng cố gắng phê phán hệ thống.
- + Người đánh giá có thể chỉ rõ điểm nhầm lẫn ở thời điểm xảy ra và cực đại hoá hiệu quả..

Thế thức phân tích kết quả

Có nhiều cách ghi lại hành động người dùng :

i) Bút và giấy: Đây là phương pháp rất nguyên thủy, song rẻ và cho phép người phân tích ghi lại các giải nghĩa và các sự kiện ở ngoài như nó đang xảy ra, Tuy nhiên, theo cách thức này khó có thể nhận được các thông tin chi tiết khi mà nó bị hạn chế bởi tốc độ chép. Để làm tăng tốc độ ghi, sơ đồ mã hoá các sự kiện thường xuyên xảy ra có thể được nghiên cứu trước. Tất nhiên, thay vì ghi chép bằng bút và giấy, ta có thể dùng máy tính xách tay.

ii) Ghi âm: Đây là phương pháp hữu ích nếu người dùng hoạt động theo kiểu "think aloud". Tuy nhiên, có thể sẽ khó để ghi đủ thông tin nhằm xác định chính xác các hành động sẽ phân tích sau này và cũng sẽ khó khăn khi sánh thông tin ghi được với thông tin thu được theo một thế thức khác, thí dụ như viết tay.

iii) Ghi hình: Phương pháp này thể hiện rõ ưu điểm là chúng ta có thể thấy chủ thể đang được thực hiện khi mà chủ thể nằm trong tầm ngắm của camera. Việc lựa chọn chính xác vị trí và góc quay của camera cho phép thu nhận được các thông tin một cách chi tiết. Thí dụ với các nhiệm vụ đơn người dùng trên máy tính, dùng 2 camera, một quan sát màn hình máy tính, một quan sát mặt và tay người dùng sẽ cho ta nhiều thông tin chi tiết hơn..

iv) Nhật ký máy tính: Điều này là tương đối dễ để có được một cách tự động một hệ thống nhằm ghi lại các hành động của người dùng ở mức độ phím nhấn nếu điều này đã được xem xét trong lúc thiết kế. Phương pháp này sẽ khó hơn đối với phần mềm có bản quyền bởi vì khi đó mã nguồn sẽ không có. Tuy rằng, nhật ký máy tính chỉ cho chúng ta biết người dùng đang làm gì trên hệ thống, song cũng có ích trong một số trường hợp.

v) Công cụ phân tích thế thức tự động: Thế thức phân tích dù là quan sát, quay video, hay nhật ký máy tính đều là các công việc tẻ nhạt, làm bằng tay. Một giải pháp có thể cho vấn đề này là cung cấp một công cụ phân tích tự động nhằm hỗ trợ nhiệm vụ. Tuy rằng

phần lớn các hệ thống kiểu này là thực nghiệm, song nó cũng cung cấp một phương tiện để hiệu chỉnh và chú thích cho việc quan sát, quay video hay ghi nhật ký và đồng bộ hoá chúng cho một phân tích chi tiết.

9.4.3. Kỹ thuật hỏi đáp

Hỏi đáp là kỹ thuật kém hình thức hơn so với các kỹ thuật thực nghiệm, song cũng có ích trong liệt kê một cách chi tiết quan điểm của người dùng về hệ thống. Kỹ thuật này có thể dùng trong đánh giá và trong sưu tập thông tin về yêu cầu của người dùng và nhiệm vụ. Ưu điểm chính của các kỹ thuật này là có thể nhận được quan điểm của người dùng một cách trực tiếp và bộc lộ một số mô thức mà người thiết kế chưa để ý đến. Hơn nữa, các kỹ thuật này là khá đơn giản và rõ đối với người quản trị. Tuy nhiên, thông tin thu được cần phải có chú ý và phải được chọn lọc. Người ta thường sử dụng hai kỹ thuật chủ yếu là phỏng vấn và hỏi.

Phỏng vấn là kỹ thuật nhằm thu thập thông tin một cách trực tiếp và có cấu trúc về kinh nghiệm của người dùng với một hệ thống tương tác. Kỹ thuật này tỏ ra khá mềm dẻo khi mà các câu hỏi có thể đa dạng, phù hợp với ngữ cảnh. Một cuộc phỏng vấn thường theo cách tiếp cận trên xuống và bắt đầu với một câu hỏi chung về nhiệm vụ và chi tiết dần theo các mức độ như tại sao như vậy, hay cần cái gì,... Phỏng vấn có thể hiệu quả đối với đánh giá mức độ cao, nhất là việc liệt kê các thông tin về sở thích, ấn tượng và thái độ của người dùng. Nó cũng có thể phát hiện các vấn đề mà người thiết kế không tiên liệu được hoặc cái không xảy ra khi quan sát.

Để phỏng vấn có hiệu quả như mong muốn cần có kế hoạch trước, và cần xác định một số thông số như số câu hỏi chính, các gợi ý. Người phỏng vấn cũng cần được lựa chọn cho phù hợp với mỗi người dùng để thu được nhiều thông tin xác đáng.

Hỏi và giám sát là kỹ thuật thứ hai được sử dụng trong nhóm này. Hiển nhiên là kỹ thuật này kém mềm dẻo hơn so với kỹ thuật trên. Tuy nhiên, nó có thể được dùng cho nhóm các chủ thể rộng hơn, ít tốn thời gian hơn và có thể được phân tích một cách chặt chẽ. Người đánh giá cũng không bị cuốn hút vào vòng hỏi đáp nếu việc thiết kế hỏi được chuẩn bị tốt. Trong kỹ thuật này, người ta có thể sử dụng rất nhiều kiểu câu hỏi: câu hỏi dạng tổng quát, dạng đóng hay dạng mở, dạng thang bậc hay dạng đa lựa chọn.

Câu hỏi dạng tổng quát nhằm thiết lập tri thức nền cho người dùng và vị trí của người đó trong tập người dùng. Nó thường bao gồm các vấn đề về tuổi tác, giới tính, các điều quan tâm,... Nó cũng có thể bao gồm các câu hỏi về kinh nghiệm dùng máy tính và có thể được diễn đạt như câu hỏi đóng mở, hay đa lựa chọn.

Câu hỏi đóng mở yêu cầu người dùng cung cấp chính kiến của mình không có gợi ý về một vấn đề, thí dụ như: "bạn có thể gợi ý gì về việc cải tiến giao tiếp này"? Nó rất có ích để thu thập các thông tin chủ quan nhưng rất khó thêm vào.

Câu hỏi dạng thang bậc yêu cầu người dùng phán xử một phát biểu đặc biệt về một tỉ lệ số tương ứng với một số đo về sự chấp thuận hay không chấp thuận với phát biểu. Thí dụ như dạng sau:

Chương 9: Các kỹ thuật đánh giá giao tiếp người dùng

Dễ dàng khôi phục do lỗi

Không đồng ý: 1 2 3 4 5 đồng ý.

Thường người ta hay dùng các độ chia từ 1 đến 5 hay từ 1 đến 7.

Câu hỏi dạng đa lựa chọn là dạng cũng khá hay dùng. Một danh sách các lựa chọn được liệt kê và người dùng - người được hỏi sẽ lựa chọn một theo chính kiến của mình. Thí dụ, đề hỏi về việc sử dụng trợ giúp, người đánh giá có thể đưa ra câu hỏi kiểu sau:

Bạn có thường sử dụng trợ giúp của hệ thống?

- Dùng tài liệu hướng dẫn
- Dùng trợ giúp ngữ cảnh
- Hỏi đồng nghiệp

Hoặc dạng sau đây:

Bạn có thể sử dụng các lệnh của hệ Soạn thảo văn bản?

	Yes	No	Don't know
Duplicate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paste	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

9.5. Lựa chọn phương pháp

Như chúng ta đã thấy, có rất nhiều kỹ thuật đánh giá và các kỹ thuật này được dùng trong nhiều giai đoạn của vòng đời phát triển. Mỗi phương pháp có những ưu điểm và nhược điểm riêng. Vậy, việc lựa chọn phương pháp nào cho phù hợp với nhu cầu của chúng ta là một việc làm cần thiết và không dễ. Do vậy, nhiều yếu tố cần được xem xét khi lựa chọn các kỹ thuật đánh giá. Người ta thường quan tâm đến các yếu tố sau:

- Các giai đoạn trong vòng đời mà đánh giá thực hiện
- Hình thức đánh giá
- Mức độ chủ quan hay khách quan của kỹ thuật
- Kiểu số đo cung cấp
- Thông tin cung cấp
- Mức độ đan xen
- Tài nguyên yêu cầu

Chúng ta sẽ phân tích một số yếu tố sau:

Chương 9: Các kỹ thuật đánh giá giao tiếp người dùng

i) Thiết kế ngược với cài đặt

Chúng ta mong muốn đánh giá cũng được bao hàm trong giai đoạn thiết kế vì những lợi ích mà nó mang lại như đã trình bày trong phần trên. Đánh giá trong giai đoạn thiết kế có khuynh hướng chỉ liên quan đến các chuyên gia và mang tính phân tích, trong khi đó đánh giá cài đặt xem xét người dùng như là chủ thể và là thực nghiệm.

Đánh giá sớm, dù đó là thiết kế hay mẫu thử hay các kịch bản mỗi sẽ mang lại sự chấp thuận nhiều nhất khi mà các vấn đề được phát hiện có thể được khắc phục ngay trong giai đoạn này

ii) Phòng thí nghiệm ngược với thực nghiệm

Đánh giá trong phòng thí nghiệm cho phép kiểm tra thực nghiệm và quan sát không cần có mặt người dùng, song không tự nhiên. Đánh giá thực nghiệm khắc phục điểm yếu trên. Lý tưởng là tiến hành cả hai kiểu trong giai đoạn thiết kế.

iii) Khách quan ngược với chủ quan

Nhiều kỹ thuật mang tính chủ quan như Walkthrough hay Think aloud dựa vào tri thức và kinh nghiệm của các nhà đánh giá, người phải nhận thức được và phải hiểu cái người dùng sẽ làm..

iv) Định lượng ngược với định tính

Kiểu số đo cung cấp bởi các kỹ thuật đánh giá là một yếu tố quan trọng. Hai loại số đo được quan tâm là số đo định tính và số đo định lượng. Số đo định lượng thường là số và dễ phân tích khi dùng các kỹ thuật như thống kê; số đo định tính là phi số và khó phân tích nhưng có thể cung cấp các chi tiết quan trọng mà không thể xác định bằng số.

Định lượng hay định tính có liên quan đến tính chủ quan hoặc khách quan của kỹ thuật. Kỹ thuật chủ quan có khuynh hướng cung cấp số đo định tính; kỹ thuật khách quan cung cấp các số đo định lượng. Có thể ánh xạ giữa hai kiểu số đo, thí dụ như đánh giá kiểu hỏi đáp là định tính, song có thể định lượng theo tỉ lệ.

v) Thông tin cung cấp

Mức độ thông tin yêu cầu cho đánh giá là khá đa dạng.

vi) Đáp ứng tức thời

Một số kỹ thuật như Think aloud ghi lại hành vi của người dùng tại thời điểm tương tác. Một số kỹ thuật khác như Walkthrough lại liên quan đến tập sự kiện của người dùng. Tuy nhiên các kỹ thuật tức thời có thể là một vấn đề cần phải tranh cãi khi mà quá trình đo đếm có thể làm tổn hại đến cách mà người dùng làm việc.

vii) Tính xâm phạm

Một số kỹ thuật đặc biệt là các kỹ thuật cung cấp số đo trực tiếp có ảnh hưởng đến

Chương 9: Các kỹ thuật đánh giá giao tiếp người dùng

cách thức ứng xử của người dùng. Các hành động nhạy cảm của người đánh giá có thể hỗ trợ nhằm làm giảm hành vi nguy hiểm của người dùng, tuy nhiên không thể loại bỏ tất cùng một lúc. Phần lớn các kỹ thuật đánh giá tức thì có tính xâm phạm trừ nhật ký hệ thống tự động.

viii) Tài nguyên

Yếu tố cuối cùng cần phải xem xét khi lựa chọn phương pháp đánh giá đó là sự sẵn có của tài nguyên. Tài nguyên cần xem xét bao gồm: thiết bị, thời gian, tiền bạc, kinh nghiệm chuyên gia, ngữ cảnh. Một số quyết định bị bắt buộc do hạn chế về tài nguyên thí dụ như phải dùng camera để đánh giá song lại không có. Khi mà thời gian và tiền bạc bị hạn chế cũng sẽ ảnh hưởng đến quyết định. Trong những tình huống như vậy, người đánh giá phải quyết định đánh giá nào sẽ được chọn để có hiệu quả, thông tin nào có ích cho hệ thống trong điều kiện xem xét.

Cũng cần chú ý là một số kỹ thuật liên quan chặt chẽ với kinh nghiệm của người đánh giá hơn các kỹ thuật khác. Nếu kinh nghiệm của người đánh giá bị hạn chế thì nên dùng các kỹ thuật heuristic hơn là các kỹ thuật phân tích. Ngữ cảnh cũng có thể ảnh hưởng tới kết quả đánh giá. Bởi những lý do thực tế, khó có thể hiểu được khuynh hướng của người dùng hệ thống (nếu đó là một hệ thống khái quát) hay không thể thực hiện được đánh giá hệ thống trong môi trường mong muốn (thí dụ đó là trên trạm vũ trụ hoặc một hệ thống phòng thủ). Trong những tình huống như thế, mô phỏng là tốt hơn cả.

Phân loại các kỹ thuật đánh giá

Mỗi họ phương pháp và mỗi phương pháp trong một họ cũng có những ưu khuyết và cách dùng khác nhau. Phương pháp thực nghiệm xem ra có vẻ hợp lý nhất trong đánh giá tính tiện dụng của giao tiếp người - máy vì nó có sự hiện diện của người dùng. Tuy nhiên, trong thực tế lại ít dùng vì yếu tố thời gian, kinh nghiệm hoặc không có thói quen cho việc đó. Các phương pháp giám sát, ngược lại không đòi hỏi chuẩn bị quá nhiều hoặc kinh nghiệm và phương pháp này tỏ ra dễ áp dụng và tích hợp trong quá trình phát triển. Phương pháp giám sát cũng không thể thay được phương pháp thực nghiệm mà chỉ bổ sung một quá trình thiết kế lặp nhằm tìm ra cơ hội phát hiện các vấn đề có tính tiềm ẩn của tính tiện dụng.

Vấn đề quan trọng ở đây là sau khi đã phân loại, có các đặc tả kiểm tra tính dùng được và cho một tập các phương pháp cần lựa chọn ra tập các phương pháp tối ưu nhất. Một số kết quả nghiên cứu tại Đại học Tổng hợp Helsinki Phần Lan (bảng 9.1) là một ví dụ tham khảo cho sự phân loại và lựa chọn các kỹ thuật đánh giá.

Chương 9: Các kỹ thuật đánh giá giao tiếp người dùng

Bảng 9.1. Lựa chọn các phương pháp đánh giá

	Mô thức	Phương pháp khuyến cáo
Mục đích của đánh giá	Khởi tạo ý tưởng mới	Đánh giá tính tiện dụng, giám sát ngữ cảnh
	So sánh đánh giá	Đánh giá tính tiện dụng, giám sát nhận thức, GOMS
	Đánh giá chấp thuận	Đánh giá tính tiện dụng, giám sát ngữ cảnh, rà soát lại chuẩn
Kiểu sản phẩm	Phần mềm	Đánh giá tính tiện dụng, giám sát ngữ cảnh, rà soát lại chuẩn, heuristic, giám sát trực quan
	Sản phẩm thông minh	Đánh giá tính tiện dụng, heuristic, giám sát trực quan, giám sát ngữ cảnh, giám sát không hình thức
Chức năng mẫu thử	Đặc tả	Giám sát
	Mẫu thử giấy	Giám sát tính tiện dụng đa chiều
	Mẫu thử chức năng	Đánh giá tính tiện dụng
Tài nguyên hiện có	Tiền và thời gian	<i>Tối thiểu:</i> Giám sát. <i>Khiêm tốn:</i> Giám sát tính tiện dụng đa chiều. <i>Tối đa:</i> Đánh giá tính tiện dụng.
	Các chuyên gia	Giám sát nếu có các chuyên gia
Các thuộc tính được đánh giá	Tính dễ học	Đánh giá tính tiện dụng, giám sát không hình
	Tính dễ nhớ	Đánh giá tính tiện dụng
	Tính hiệu quả	Đánh giá tính tiện dụng, GOMS
	Lỗi	Mọi phương pháp thích hợp
	Đáp ứng	Đánh giá tính tiện dụng

Bạn đọc quan tâm hơn về cách phân loại và lựa chọn phương pháp, xin mời đọc tài liệu tham khảo [19].

10

MULTIMEDIA VÀ WEB

Nội dung

10.1. Tổng quan về multimedia và Web

10.1.1. Multimedia

10.1.2. Web

10.2. Các đặc điểm của Web

10.2.1. Người dùng

10.2.2. Thông tin và kiến trúc trang Web

10.3. Các tiêu chí thiết kế và đánh giá trang Web

10.3.1. Mười lỗi phổ biến

10.3.2. Các tiêu chí thiết kế

10.4. Thiết kế Web

10.5. Một số xu hướng hiện nay

Hiện nay, mạng Internet và thương mại điện tử đang trên đà phát triển mạnh mẽ. Internet đang trở thành một môi trường làm việc không thể thiếu đối với các cá nhân cũng như với các tổ chức, doanh nghiệp.

Vi vậy, việc xây dựng các ứng dụng trên nền Internet ngày càng có vai trò quan trọng không kém gì các ứng dụng truyền thống thông thường. Một trong số đó là các ứng dụng trên nền Web (Web - based Application).

Việc xây dựng một ứng dụng Web có khá nhiều khác biệt so với các ứng dụng thông thường. Nếu như một hệ thống Web nào đó được xây dựng không tốt, không tạo được sự tương tác thích hợp với người dùng thì sẽ không thể chuyển tải thông tin một cách hiệu quả, dẫn đến sự vô ích của hệ thống đó. Ngược lại, nếu một trang Web có tính tương tác tốt giữa người dùng - đối tượng sử dụng Web với máy tính, sẽ tạo nên sự hấp dẫn lôi cuốn. Do có sự khác biệt giữa ứng dụng thông thường với ứng dụng Web, việc đề ra các tiêu chí cho thiết kế cũng như đánh giá tính tiện dụng là một đòi hỏi quan trọng. Mục đích của chương này là nhằm tìm hiểu các đặc điểm khác biệt của Web, trên cơ sở đó đưa ra các tiêu chí cho việc thiết kế, đánh giá và xem xét quá trình thiết kế Web - một quá trình thiết kế đa phương tiện chung, song cũng là một quá trình đặc biệt. Tuy nhiên, trước hết cần tìm hiểu một số khái niệm quan trọng được dùng trong công nghệ Web như siêu văn bản (hypertext), siêu liên kết (hyperlink), đa phương tiện (multimedia),... Đó là nền tảng của công nghệ Web.

10.1. Tổng quan về multimedia và Web

Web ra đời vào cuối năm 1989 tại viện nghiên cứu hạt nhân châu Âu ở Gionevơ, Thụy Sĩ (European Laboratory for Particle Physics in Geneva) do Time Berners Lee đứng đầu cùng với các cộng sự của mình, với ý tưởng tạo ra một loại siêu văn bản thông minh – HyperText. Dần dần, cùng với tính ưu việt của mình, cộng với các thay đổi về công nghệ, các trình duyệt Web nổi tiếng ra đời như Mosaic, Netscape, Internet Explorer, đã làm cho việc sử dụng Web trở nên dễ dàng hơn nhiều và Web đã trở thành phổ biến. Muốn có các trang Web đương nhiên phải có các nhà thiết kế và phục vụ cho các mục đích riêng của mình như nghiên cứu, đào tạo, trao đổi thông tin, quảng cáo,...

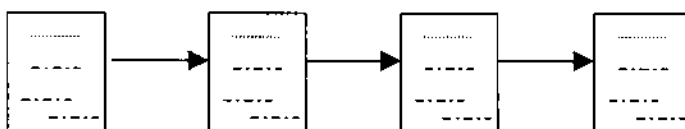
10.1.1. Siêu văn bản, siêu liên kết, multimedia

Ngày nay những khái niệm này đã khá quen thuộc với phần lớn mọi người trong chúng ta, không chỉ riêng những người làm công nghệ thông tin. Tuy nhiên, khoảng hơn mười năm trước thôi những thuật ngữ này có lẽ còn chưa xuất hiện chứ không muốn nói đến khái niệm.

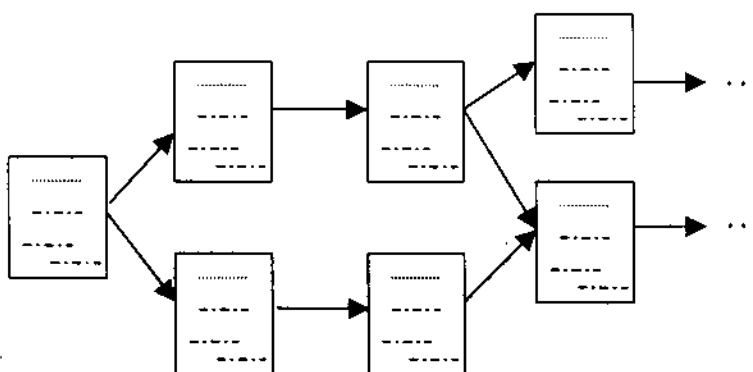
Khái niệm Siêu văn bản do nhà tin học Ted Nielson đưa ra năm 1965 trong bài viết “Computer Dreams”, bàn về ý tưởng tạo ra các máy tính có tư duy như con người. Nhưng người thực hiện ý tưởng ấy lại là Tim Berners Lee vào khoảng những năm 80 và đến năm 1989, Tim và các cộng sự của mình đã thực hiện thành công dự án lưu trữ siêu văn bản trên máy tính để dễ dàng tìm các tài liệu liên quan.

Thực tế, không có một định nghĩa cụ thể về siêu văn bản. Người ta cho rằng một siêu văn bản được tổ chức theo kiểu lưới, mỗi ô chứa một đoạn (trang) của văn bản. Các đoạn này sẽ được liên kết với nhau theo nhu cầu sử dụng. Như vậy, một hệ thống siêu văn

bản sẽ bao gồm một số các **trang** và một tập **các liên kết (hyperlink)**. Chính các liên kết này cho phép tập hợp các trang lại với nhau. Việc liên kết có thể nối bất kỳ một trang này vào một trang khác không phụ thuộc vào tính vật lý của trang: có thể là các phần khác nhau trong một văn bản vật lý, có thể là các văn bản khác nhau trên cùng một máy tính hay trên mạng internet. Hơn nữa, một trang có thể có nhiều liên kết tới các trang khác. Do vậy, một siêu văn bản không phải là một kiểu tuyến tính từ một trang bắt đầu và cứ lần lượt mà nó có thể đi theo nhiều hướng khác nhau: cái thì đã kết thúc, cái lại quay trở lại một phần nào đó của văn bản. Sự khác nhau giữa một văn bản tuần tự và một siêu văn bản được minh họa trên hình 10.1.



Hình 10.1a. Văn bản tuyến tính.



Hình 10.1b. Siêu văn bản.

Một cách đặc trưng, một hệ thống siêu văn bản bao hàm trong nó ngoài văn bản còn có các sơ đồ, hình ảnh và các phương tiện (media) khác. Một hệ thống như vậy thường được biết đến như đa phương tiện (multimedia), hay siêu phương tiện (hypermedia). Đa phương tiện, siêu phương tiện hay siêu văn bản là các tên gọi khác nhau, song trong thực tế lại được dùng không khác nhau.

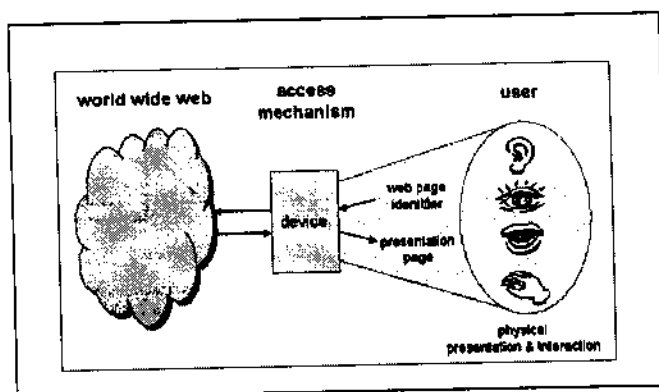
Chú ý rằng đa phương tiện hay siêu phương tiện không chỉ dành cho ứng dụng Web mà còn trong nhiều các ứng dụng khác như xây dựng các công cụ cho quá trình phát triển phần mềm (tạo mẫu thử, đánh giá thiết kế,...), các hệ thống đào tạo từ xa, v.v. Tuy rằng, đây không phải điều quan tâm chính của chúng ta, song có lẽ tốt hơn nếu chúng ta đi qua nó. Nhiều công cụ nổi tiếng được dùng trong sản xuất phần mềm như Hypercard, Revolution, Flash Get của Marco Media là những hệ thống siêu phương tiện điển hình.

Hypercard là một công cụ được dùng khá rộng rãi trong giới máy tính Macintosh. Revolution là một sản phẩm của hãng Runtime Revolution Team phát triển cho các dòng máy PC-IBM với nhiều phiên bản khác nhau ((xem 4.5.4 chương 4).

10.1.2. Web

Web là một hệ thống siêu phương tiện toàn cầu nổi tiếng nhất hiện nay. Web bao gồm một tập các giao thức được xây dựng trên đỉnh Internet. Chính nhờ các giao thức này, các tài liệu đa phương tiện được tạo ra và được đọc từ bất kỳ máy tính nối mạng nào trên thế giới. Web hỗ trợ siêu văn bản, hình ảnh, âm thanh, phim và sử dụng ngôn ngữ đặc biệt là ngôn ngữ đánh dấu siêu văn bản – HTML (HyperText Markup Language). HTML cho phép nhúng các siêu liên kết, hình ảnh, âm thanh và phim vào văn bản, đồng thời cung cấp một số tiện ích để miêu tả cách mà các thành phần này được bố trí, đó chính là nhờ các thẻ (tag).

Giao thức nổi tiếng, cho phép đọc nhanh các tài liệu loại này là HTTP (HyperText Transfer Protocol). HTTP là giao thức truyền thông cộng thêm khả năng phân tích các liên kết để truy nhập đến các tư liệu liên quan trên Internet. Các trang Web ngoài việc thể hiện bằng ngôn ngữ HTML còn có thể được nhúng thêm các Applet hay các Script phục vụ cho nhu cầu truy nhập động.



Hình 10.2. Cơ chế tương tác của người dùng với Web qua trình duyệt.

Các trang Web là các trang html hay mở rộng hơn là các trang asp, jsp được giải nghĩa bằng các trình duyệt (Web browser). Các trình duyệt này có rất nhiều (10,1) và hoạt động theo nhiều cách khác nhau, hỗ trợ theo nhiều mức độ chức năng khác nhau. Vì vậy, cùng một trang Web được duyệt qua một trình duyệt có thể sẽ rất khác nhau trên các máy khác nhau. Web không yêu cầu các khả năng đa phương tiện đặc biệt của máy mà trình duyệt thực hiện. Thí dụ như nếu máy không có card âm thanh, chúng ta không nghe thấy âm thanh, song trình duyệt vẫn hiện đúng các thông tin.

10.2. Các đặc điểm của Web

10.2.1. Người dùng

Khi thiết kế trang Web rất cần quan tâm tới đối tượng người dùng. Đối tượng dùng Web rất đa dạng về nghề nghiệp, giới tính, tuổi tác cũng như mục đích truy nhập. Và có lẽ không có phần mềm nào có người sử dụng đa dạng như Web.

Điều thứ hai, người dùng không thích các kết nối bị đổ vỡ (Linkrot), mà điều này cũng rất khó tránh. Các thống kê cho rằng có 0,6% kết nối bị đổ vỡ và 23% các trang Web có chứa ít nhất một kết nối không thực hiện được.

Thứ ba là người dùng không thích cuộn trang và ít khi đọc kỹ nội dung trang Web. Các nghiên cứu của Morkes và Nielsen (1997) cho thấy 79% số người truy cập chỉ đọc lướt và chỉ có 16% là đọc kỹ. Thường người dùng chỉ thích đọc các mục làm nổi (highlight), ví dụ như các tiêu đề, các kết nối, v.v. Vì thế, người dùng thích mỗi đoạn có một chủ đề và họ có thể bỏ qua nếu nó không hấp dẫn. Một lý do nữa là đọc trên màn hình chậm hơn trên giấy, tốc độ đọc chỉ bằng 25% so với đọc trên giấy. Do vậy, khi đưa nội dung lên trang Web phải tính đếm làm sao số từ chỉ khoảng 50% của trang khi in ra giấy.

Đặc điểm thứ tư, cũng theo Nielsen thì người dùng thích sử dụng công cụ tìm kiếm hơn là các kết nối. Số người sử dụng công cụ tìm kiếm là 50%, trong đó dùng kết nối là 30%, kết hợp cả hai loại là 20%. Tuy nhiên, người dùng thường không biết các qui tắc tìm kiếm, hay nhập sai các từ khoá và thường không thay đổi các tuỳ chọn trong tìm kiếm. Với các trang Web khác nhau, các qui định về cú pháp cũng khác nhau.

Đặc điểm thứ năm là người dùng có thể “viếng thăm” Web bằng nhiều con đường khác nhau: thông qua tìm kiếm, thông qua trang chủ hay thông qua các liên kết để đến thăm một trang bất kỳ của một Website chứ không nhất thiết từ trang chủ, theo trình tự như ý đồ của người thiết kế trang Web. Vì trang Web hầu như không có điểm dừng nên người dùng cũng phải mò mẫm khá lâu trước khi tìm được thông tin cần thiết nếu chỉ đơn thuần đi theo cá đường liên kết.

10.2.2. Thông tin và Kiến trúc trang Web

Trước hết phải nói rằng, mỗi trang Web chứa một lượng thông tin khổng lồ và hầu như không có kết thúc (hình 10.1.b). Nhờ có các siêu liên kết (hyperlinks) mà các trang Web có thể nối với nhau và điều này cho phép các trang Web không bị hạn chế về không gian. Có nghĩa là các trang Web có thể phân bố trên các site khác nhau trên Internet. Điều này trong suốt với người dùng, làm họ có cảm giác mình đang sử dụng một văn bản thống nhất. Tuy nhiên, điều bất lợi do phân bố ở nhiều site vật lý khác nhau trên Internet nên thời gian truy tìm và tải được thông tin sẽ khá chậm. Chúng ta sẽ phân tích kỹ điểm yếu này ở phần sau. Các liên kết này có thể thực hiện bởi các đoạn văn bản hay hình ảnh hay một phím lệnh (command button) nào đó.

Đặc điểm thứ hai là thông tin trên Web rất đa dạng. Như đã nói ở trên, Web là một hệ thống siêu văn bản và ngày càng có nhiều phương tiện khác được bổ sung và Web.

Thông tin hình ảnh cũng là một nét đặc trưng của Web. Người dùng Web không chỉ xem xét thông tin như cách truyền thống trên giấy mà theo cách riêng, sử dụng các ấn dụ, hình ảnh, các siêu liên kết. Hình ảnh cùng với giao tiếp đồ họa làm cho trang Web sinh động, đẹp. Web sử dụng rất nhiều dạng ảnh như: BMP, GIF, JPG, XBM, PNG,...

Ngoài ảnh tĩnh, Web còn sử dụng khả năng hoạt hình (animation), video. Hoạt hình là sự tạo ảo giác về sự chuyển động, có thể là trong một chương trình máy tính hay trên màn ảnh bằng cách gửi một loạt hình ảnh liên tiếp biểu diễn những thay đổi vị trí từ từ của các đối tượng. Việc hoạt hình được thực hiện bằng cách vẽ các khung hình chính (do các họa sỹ thực hiện) và máy tính morphing các khung hình này để tạo ra các khung hình trung gian (in-betwens). Hầu hết các hoạt hình hiện nay trên máy tính đều là 3 chiều sử dụng mô hình hướng sự kiện. Hoạt hình được đưa vào Web nhờ các ảnh Gif hoặc các đối tượng của FLASH. Việc sử dụng hoạt hình làm cho trang Web thêm sống động, gây ấn tượng cho người dùng hơn hình ảnh thông thường nhưng cũng làm tăng kích thước trang Web và làm tăng thời gian tải trang.

Trong môi trường Web, thị giác không phải là cách giao tiếp duy nhất. Công nghệ Web có thể thực hiện được việc giao tiếp âm thanh nhờ các tệp audio. Các dạng tệp audio phổ biến như; mid hay midi, wav, aif, mp3,... Với việc giao tiếp qua âm thanh, người dùng có thể nghe các bản tin, ca nhạc mà không cần dán mắt vào màn hình. Hai trình duyệt miễn phí có thể xử lý âm thanh là RealPlayer của RealNetwork và MicroSoft Media Player của MicroSoft.

Việc tích hợp các tệp video (chủ yếu dạng MPEG) làm cho người dùng Web có thể xem phim trực tuyến hay tải một bộ phim từ internet để xem trong chế độ offline. Với việc tích hợp video, Web đã làm tăng khả năng của mình như: trong đào tạo từ xa, các học viên có thể tham gia học trực tiếp qua mạng, các doanh nghiệp có thể dựng một đoạn phim, quảng cáo,... Trở ngại lớn nhất trong việc phát triển công nghệ video đó là tốc độ và băng thông đường truyền..

10.3 Các tiêu chí thiết kế và đánh giá trang Web

Với các đặc điểm người dùng và các đặc điểm về hình thức cũng như sự đa dạng về nội dung, để thiết kế một trang Web mang tính tiện dụng cao cần đặt ra các tiêu chí cụ thể. Các tiêu chí này không chỉ dùng trong giai đoạn thiết kế mà còn để đánh giá trang Web. Trước tiên chúng ta sẽ xem xét các lỗi hay gặp hay đó chính là thói quen của các nhà thiết kế Web, sau đó sẽ định ra tiêu chí thiết kế.

10.3.1. Mười lỗi phổ biến

Theo Nielsen và một số các chuyên gia, có 10 lỗi phổ biến thường gặp khi thiết kế các trang Web [10]:

1. Thời gian tải khá lâu, nhất là với các trang Web sử dụng nhiều hình ảnh
2. Các thông tin thường không được cập nhật
3. Vấn đề màu sắc và sự thống nhất các liên kết
4. Thiếu sự hỗ trợ của người dùng trong khi khám phá các trang Web
5. Trang Web thường quá dài, nhiều chữ
6. Các trang Web thường bị cô lập (orphan page), không có trang chủ liên kết
7. Các liên kết, địa chỉ phức tạp, dài và khó nhớ
8. Các hình động hay hoạt hình làm rối mắt và giảm tốc độ
9. Hiện tượng chày máu chất xám vô ích
10. Việc sử dụng khung không hiệu quả

10.3.2. Các tiêu chí thiết kế và đánh giá trang Web

Việc thiết kế một trang Web mang tính tương tác rất cao và việc thiết kế cần đi vào chi tiết cho từng trang. Muốn thiết kế tốt phải đảm bảo các tiêu chí sau:

1. Thiết kế trực quan (Visual Design & layout)
2. Sử dụng không gian màn hình (Use of Screen real estate)
3. Thời gian tải
4. Các kỹ thuật HTML để thể hiện giao diện
5. Thao tác giữa các phần hay môi trường người dùng

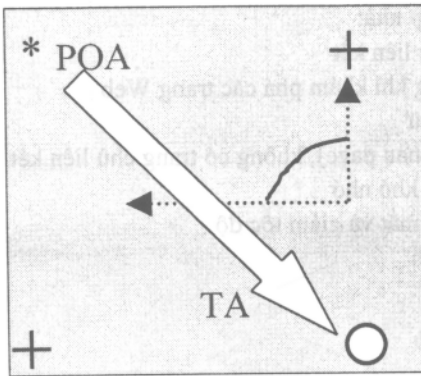
10.3.2.1. Thiết kế trực quan

Các trang Web thiết kế tốt bao giờ cũng hướng tới người dùng, đảm bảo tương tác tốt nhất, tránh gây phiền hà và cung cấp các phương tiện để họ duyệt trang. Để làm điều đó cần quan tâm tới:

a) Sự di chuyển mắt khi nhìn của người (eye flow)

Người ta thấy rằng có sự liên hệ mật thiết giữa sự di chuyển của mắt trong khi duyệt và quá trình xử lý thông tin thu nhận được qua việc dịch chuyển này. Thiết kế tốt là dựa vào luồng dịch chuyển. Nếu người dùng càng phải di chuyển mắt nhiều để quan sát các đối tượng trên trang Web, họ càng thu lượm được ít thông tin (Duff & Mohler, 1996). Nếu không quan tâm đến điều này, người dùng không thể đọc được thông tin mà chỉ lướt qua trang Web mà thôi.

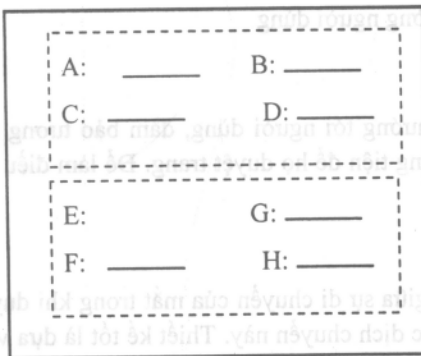
Chú ý rằng, con người có thói quen đọc từ trái qua phải và từ trên xuống dưới. Có thể hình dung điều này thông qua sơ đồ Gutenberg trong hình 10.3 dưới đây.



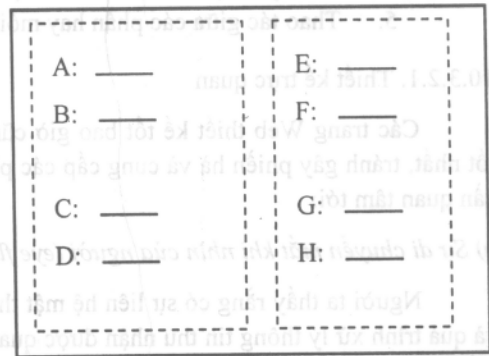
- POA (Primary Optical Area): điểm xuất phát.
- TA (terminal area): điểm kết thúc.
- Đường chấm gạch: thể hiện hướng đưa mắt ngược với hướng đưa mắt của người.
- Dấu '+': đánh dấu biên.

Hình 10.3. Lưu đồ Gutenberg.

Theo nguyên tắc trên thì việc bố trí các đối tượng như chỉ ra trong hình 10.4b là không tốt, các đối tượng bố trí trong hình 10.4a là tốt.



Hình 10.4a. Thiết kế đúng.



Hình 10.4b. Thiết kế không đúng.

Theo các chuyên gia, sử dụng ô lưới trong việc bố trí các phần tử sẽ làm giảm eye flow. Do vậy, khi thiết kế trang Web nên bố trí các phần tử theo một trật tự, tạo ra các đường ảo (không nhìn thấy, song có thể tưởng tượng qua sự bố trí các phần tử). Thí dụ như kẻ bảng hay các hộp văn bản (textbox) với đường bao có độ dày bằng không rồi xếp các phần tử vào đó. Việc sử dụng ô lưới giúp cho người dùng có thể đoán được hướng di chuyển để đọc nội dung trang Web cũng như giúp nhà thiết kế kiểm tra sự sắp xếp và thay thế các phần tử trong trang Web khi cần thiết. Hình 10.5 dưới đây minh họa việc bố trí theo lưới của một trang Web.

YAHOO! Mail Help: - Yahoo!

Sign up for your Yahoo! ID with Mail Already have an ID? [Sign In](#)

Get a Yahoo! ID and password for access to Yahoo! Mail and all other personalized Yahoo! services.

Yahoo! ID: @yahoo.com

Phone:

Password:

Re-type Password:

If you forget your password, we would identify you with this information:

Security Question: [select a question to answer]

Your Answer:

Birth day: [select crs] / [select mo] / [select day] Month Day, Year

Current E-mail:

Choosing your ID
You will use this information to access Yahoo! each time. Capitalization matters for your password.

Recalling your password
This is our only clue to verify your identity. To protect your account, make sure your answer "is memorable for you but hard for others to guess!"

Hình 10.5. Trang Web kiểu form điền và lưới.

b) Cấu trúc phân cấp trực quan

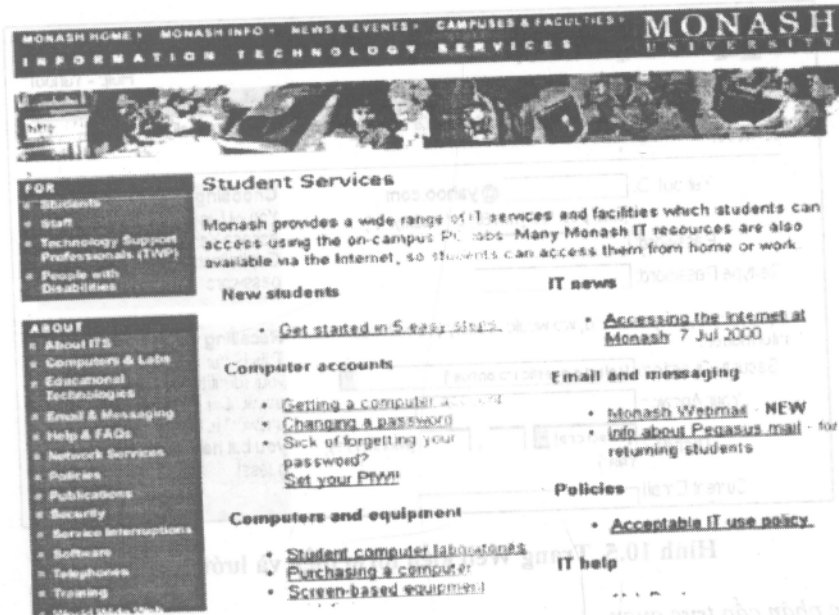
Cũng giống như eye flow, việc cung cấp cho người dùng cái nhìn trực quan về cấu trúc phân cấp của trang Web cũng ảnh hưởng đến việc thu lượm và xử lý thông tin của người dùng Web. Steve Krug đã nói về vai trò của tính phân cấp trực quan này như sau: "Một trong các cách tốt nhất làm cho trang Web của bạn được hiểu một cách nhanh chóng là phải bảo đảm các phần trên trang Web được bố trí hợp lý, rõ ràng, sinh động, phần nào liên quan đến mục nào" [23]. Bố trí trực quan được minh họa trên hình 10.6

Tính quan trọng của đối tượng phụ thuộc vào mục đích xây dựng trang Web. Để thể hiện tính quan trọng của đối tượng, người ta áp dụng các nguyên tắc sau:

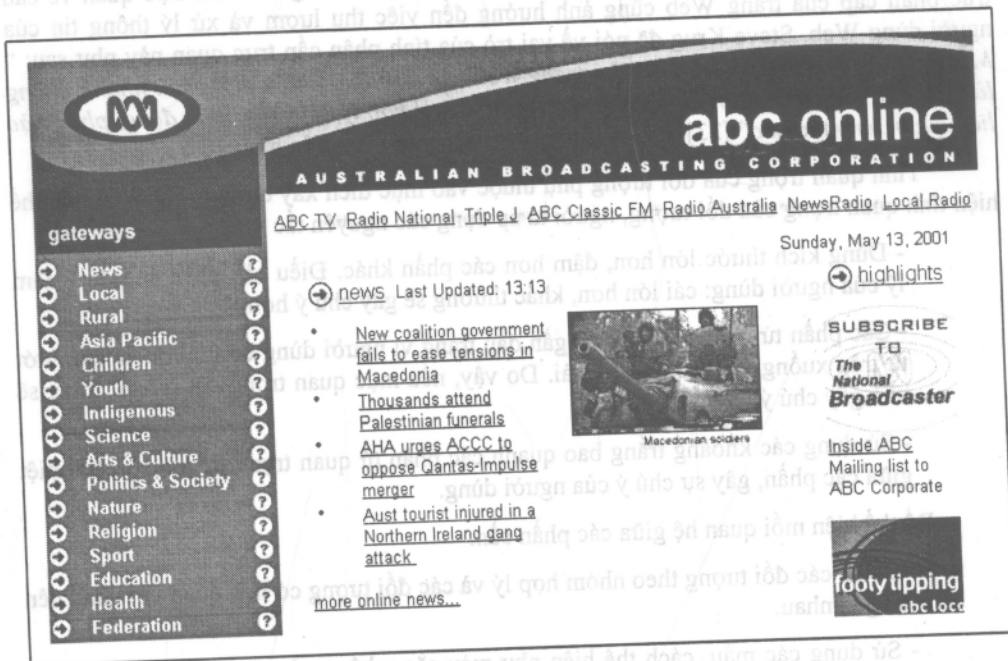
- Dùng kích thước lớn hơn, đậm hơn các phần khác. Điều này phản ánh đúng tâm lý của người dùng: cái lớn hơn, khác thường sẽ gây chú ý hơn.
- Các phần tử quan trọng đặt ở gần đầu trang vì người dùng có thói quen xem lướt từ trên xuống và từ trái qua phải. Do vậy, nếu mục quan trọng đặt ở cuối trang sẽ khó gây chú ý cho người dùng.
- Sử dụng các khoảng trắng bao quanh các phần tử quan trọng để tạo sự cách biệt giữa các phần, gây sự chú ý của người dùng.

Để thể hiện mối quan hệ giữa các phần cần:

- Bố trí các đối tượng theo nhóm hợp lý và các đối tượng có liên quan với nhau nên đặt gần nhau.
- Sử dụng các mẫu, cách thể hiện như màu sắc, phông và cỡ chữ như nhau (hình 10.7).



Hình 10.6. Cấu trúc phân cấp trực quan.



Hình 10.7. Cấu trúc phân cấp trực quan.

Chương 10: Multimedia và Web

Với cách thiết kế như vậy sẽ tạo nên sự tương phản giữa các đối tượng gây cảm giác phân biệt và chú ý, giúp người dùng dễ lướt trên trang Web. Việc tạo các tiêu đề nổi bật giúp cho người dùng nắm được nội dung ngay khi tiếp cận với dòng tiêu đề.

c) Sắp xếp văn bản hợp lý

Văn bản cần được sắp xếp hợp lý tạo khả năng đọc lướt. Các lựa chọn chính là căn lề văn bản. Trong các loại căn lề, căn lề trái là tốt hơn cả vì người dùng bắt đầu đọc từ trái sang. Hình 10.8 dưới đây minh họa nguyên tắc này.

<p>For some reasons that continues to elude me, a great many Web pages contain centred text.</p> <p>Centred text alignment is the mark of an amateur.</p> <p>Do you find this easy to read?</p>	<p>Right alignment, or 'ragged left' as it is sometimes called is thought to be quite trendy.</p> <p>What do you think about its impact on readability?</p>		
<p>Left alignment is not funky or groovy. In fact, it is the most commonly used text alignment. That is probably because it's much easier to read.</p> <p>What do you think?</p>	<table border="1" style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;"> <p>A right-aligned Chunk And more again. Where will this end</p> </td> <td style="width: 50%; vertical-align: top;"> <p>How about a mixed alignment? A bit of this left aligned text over there, and some ...</p> </td> </tr> </table>	<p>A right-aligned Chunk And more again. Where will this end</p>	<p>How about a mixed alignment? A bit of this left aligned text over there, and some ...</p>
<p>A right-aligned Chunk And more again. Where will this end</p>	<p>How about a mixed alignment? A bit of this left aligned text over there, and some ...</p>		

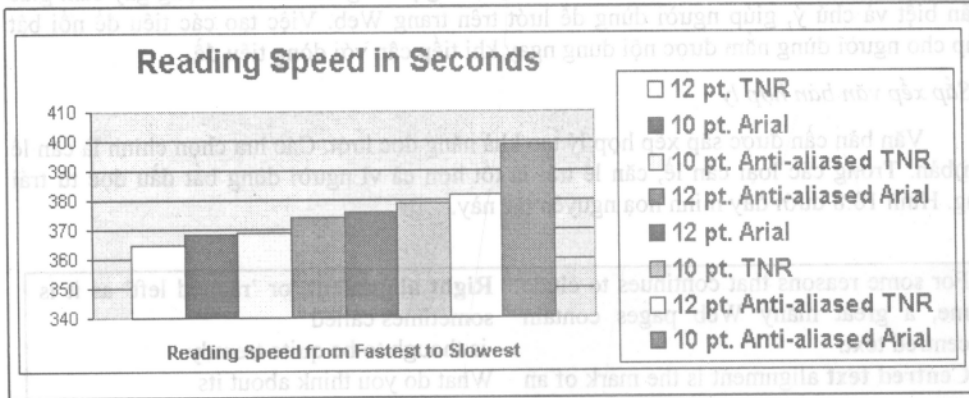
Hình 10.8. Căn lề văn bản trên trang Web.

d) Tính dễ xem (Legibility)

Các nhân tố tạo nên tính dễ xem bao gồm: cỡ chữ, phong chữ, màu sắc và sự tương phản. Các nghiên cứu trong phòng thí nghiệm (Software Usability Research Lab Wichina State University) tiến hành trên 35 người với thị lực bình thường đọc 8 mẫu văn bản khác nhau và thực nghiệm đã chỉ ra rằng với các loại phông như Arial hay Time New Roman thì cỡ chữ từ 9 đến 12 thuộc loại dễ đọc. Họ cũng khuyến cáo là tránh sử dụng toàn chữ hoa, hay dùng chữ nhấp nháy hoặc chuyển động kiểu marquee style hoặc bị phóng quá to hay thu nhỏ sẽ gây ra sự khó đọc. Kết quả thí nghiệm này được chỉ ra trên hình 10.9.

Màu sắc và độ tương phản cũng đóng vai trò quan trọng. Việc dùng màu sắc và độ tương phản hợp lý tạo cảm giác dễ chịu, ngược lại sẽ gây nên cảm giác hoa mắt hoặc căng thẳng. Về mặt tâm lý, các màu sắc đậm dễ gây ấn tượng hơn. Do vậy khi sử dụng màu sắc cần chú ý:

- Không nên dùng chữ màu xanh hay đỏ.
- Không sử dụng chữ xanh trên nền đỏ và ngược lại. Chữ đen trên nền vàng hay chữ đen trên nền trắng đều dễ đọc.



Hình 10.9. Vai trò của phông và cỡ chữ.

- Các đối tượng có màu sắc gần giống nhau không nên để cạnh nhau, cụ thể như: màu xanh lá cây gần với màu xanh nước biển, các gam màu sáng gần nhau cũng tạo nên cảm giác hỗn độn, khó chịu.

10.3.2.2. Cách sử dụng không gian màn hình

Không gian màn hình phải được sử dụng hợp lý giữa các phần, đặc biệt là phần nội dung trang Web. Các ứng dụng đặc biệt sử dụng không gian hẹp như dạng mobile phone sẽ đề cập trong một bài khác. Jakob Nielsen khuyến cáo rằng nên sử dụng 50-80% màn hình cho nội dung trang Web.

Phong cách thiết kế cũng là điều cần quan tâm. Nói chung người ta thích kiểu “above the fold” hay kiểu trang báo. Điều này có nghĩa là các thông tin quan trọng cho lên trang đầu. Người dùng không thích cuộn trang, do vậy phải chú ý thiết kế nội dung với không gian thích hợp.

Việc bố trí các phần tử trên trang Web cũng cần quan tâm. Thanh Navigation nên đặt bên trái hay phía trên cùng của trang và các liên kết tương đương nên đặt ở cuối trang. Nếu thiết lập logo thì nên đặt ở góc bên trái trên hoặc ở giữa và có liên kết với trang chủ.

10.3.2.3. Thời gian tải trang

Thời gian tải một trang Web là một tiêu chí rất quan trọng. Jakob Nielsen tổng kết các nghiên cứu của mình từ năm 1994 đến nay cho rằng [15]: người dùng luôn đề nghị tăng tốc độ cho các trang Web. Tốc độ tải tạo cho người dùng cảm giác tin cậy, không phụ thuộc thời gian và tăng hiệu năng làm việc. Sau đây là một số giới hạn về thời gian chấp nhận được:

- 0,1 giây cho các xử lý của hệ thống

Chương 10: Multimedia và Web

- 1 giây trước khi cắt ngang dòng suy nghĩ của người dùng (vấn đề độ trễ)
- 10 giây để tạo được sự chú ý của người dùng. Đó chính là thời gian giới hạn để tải về một trang Web mà người dùng không cảm thấy khó chịu.

Vậy thì nhân tố gì sẽ ảnh hưởng tới tốc độ duyệt trang? Đương nhiên, trước hết phụ thuộc vào thông lượng của máy chủ và sự kết nối máy chủ với mạng. Thứ hai là việc kết nối máy tính người dùng với mạng và một số nút cổ chai trên mạng. Các nhân tố này nằm ngoài sự kiểm soát của người thiết kế trang Web. Tuy nhiên, một số nhân tố khác lại phụ thuộc vào người thiết kế như: kích thước trang Web càng lớn thì tốc độ càng thấp hoặc cấu trúc các bảng phức tạp thì thời gian hoàn trả cũng chậm. Qua nghiên cứu và thực nghiệm, một số nguyên tắc sau phải thực hiện để đảm bảo tăng tốc độ truy nhập:

- Tránh dùng các bảng phức tạp: cần phải giảm bớt sự hoà trộn dữ liệu. Nếu các bảng lớn và phức tạp có thể chia thành nhiều bảng con.
- Sử dụng đồ hoạ và multimedia hợp lý: hạn chế tối đa dùng hình ảnh nếu có thể. Nếu phải dùng thì phải tạo kích thước tối ưu như dạng thumbai. Dùng các tệp audio, video phải hết sức cẩn thận vì các tệp này thường khá lớn.
- Giới hạn kích thước trang Web: theo kết quả công bố của Vincent Flander, 1999 khi khảo sát 50 Website hàng đầu thế giới cho thấy:

- ◆ 10 Website nhanh nhất với kích thước là 34,4 KB
- ◆ 10 Website chậm nhất với kích thước là 61,3 KB
- ◆ Kích thước trung bình là 47,8 KB

Sau đây là kích thước của một vài trang chủ của các Website nổi tiếng: Google: 13 KB, Yahoo: 19 KB, Amzon từ 65 đến 87 KB.

Nhiều nghiên cứu cũng chỉ ra phương thức kết nối mạng rất ảnh hưởng đến tốc độ. Trong đó, kết nối qua modem chậm nhất: 2 KB/ giây, 10 giây: 34 KB; mạng dịch vụ tích hợp số ISDN khá hơn: 8KB/ giây, 10 giây 150 KB.

10.3.2.4. Các kỹ thuật HTML trong giao diện

Để hỗ trợ cho việc xây dựng trang Web, các kỹ thuật cũng cần được xem xét. Ba kỹ thuật thường được quan tâm:

- Stylesheet: một số mẫu đã được định sẵn để tạo nên sự thống nhất cho toàn bộ trang Web về màu sắc, phông chữ và cỡ chữ. Tuy nhiên cũng cần tính đến việc các trình duyệt có hỗ trợ không?

- Frame: tổ chức các trang trong cùng một cửa sổ nhằm tạo nên cấu trúc Website.

- Table: bố trí các phần tử theo trật tự, tránh sự co dãn ngoài ý muốn

Việc bố trí chiều rộng cũng cần cân nhắc và thường có hai lựa chọn: cố định hay thay đổi. Việc bố trí chiều rộng thay đổi có lợi khi thay đổi độ phân giải màn hình.

10.3.2.5. Môi trường người dùng

Việc sử dụng trình duyệt Web nào là phụ thuộc người dùng. Vì vậy cần lựa chọn thiết kế thích hợp. Nhìn chung, nên lựa chọn các chuẩn hỗ trợ nhiều môi trường như HTML, XHTML v.v...

10.4. Thiết kế Web

10.4.1. Thiết kế hướng tới người sử dụng

Giao diện đồ họa người sử dụng (GUI) được thiết kế nhằm tạo thuận lợi cho người dùng có thể sử dụng kiểu điều khiển trực tiếp với máy tính của họ. Ngày nay, người sử dụng yêu cầu một mức độ hoàn hảo của mọi thiết kế giao diện đồ họa, kể cả các trang Web. Mục đích là cung cấp mọi thứ cần thiết cho người dùng, đáp ứng các mong chờ của họ, và không bao giờ đòi hỏi người dùng chỉ đơn giản là làm theo một giao diện mà đặt những cản trở không cần thiết lên con đường của họ. Ở đây, việc nghiên cứu của chúng ta về các nhu cầu và tâm lý học khách hàng là những yếu tố quyết định. Không thể thiết kế giao diện cho một người dùng mà ta không hề biết đến các yêu cầu của người đó. Chúng ta nên tạo các kịch bản mẫu cho các nhóm người dùng đang tìm kiếm thông tin trên Website của chúng ta. Liệu một người dùng có kinh nghiệm tìm một mẫu tin nhất định có được giúp đỡ hay bị cản trở bởi thiết kế của chúng ta? Một người dùng mới có bị lúng túng trước một hệ thống menu phức tạp? Thử nghiệm các thiết kế của chúng ta và nhận các phản hồi từ người dùng là phương pháp tốt nhất để tìm ra các ý tưởng thiết kế cho phép người dùng nhận được cái họ muốn từ trang Web của chúng ta.

10.4.2. Các giúp đỡ định hướng rõ ràng

Với thực tại của công nghệ Web, đa số người dùng tương tác với các trang Web bằng cách thực hiện các liên kết giữa các tài liệu. Các biểu tượng cần nhất quán, dễ hiểu, các lược đồ đồ họa thống nhất và khái quát, màn hình tổng hợp có thể cho người dùng sự tin tưởng là họ có thể tìm thấy cái họ cần mà không lãng phí thời gian.

10.4.3. Các quy tắc liên kết và điều khiển

Cung cấp một tập phong phú các điều khiển đồ họa và liên kết tương tác trong trang Web sẽ thu hút sự chú ý của người dùng xuống trang Web, quên đi các liên kết đa năng của trình duyệt và lôi cuốn họ vào nội dung. Bằng cách sử dụng các phím ấn đồng nhất, có thể dự đoán được, chúng ta đã giúp người dùng có được cảm nhận về trang Web của mình, và làm cho logic và trật tự của trang Web trở nên rõ ràng.

- **Không có trang cuối cùng (dead-end)**

Mọi trang Web nên có ít nhất một liên kết. Các trang "dead-end" - các trang không móc nối đến các trang khác trong cùng site - không chỉ là một sự thất vọng với người dùng mà còn làm mất khả năng người dùng đến với các trang khác trong trang Web của chúng ta. Các trang Web nếu đưa ra không có lời tựa đầu sẽ dẫn

đến việc độc giả thường tạo hay đi theo các liên kết thẳng đến các trang cất sâu trong cấu trúc của Website. Do vậy, họ có thể không bao giờ nhìn thấy trang chủ (Homepage) hoặc các thông tin mở đầu trên Website của chúng ta. Nếu các trang phía dưới của site không có các liên kết quay lên, về trang chủ hoặc quay lại menu, độc giả thực chất là bị loại khỏi việc truy nhập đến các phần còn lại của Web site.

- **"Quay lại", quay về trang trước và trang chủ**

Tất cả các hệ thống siêu văn bản đều chia sẻ một vấn đề chung đó là: quay lại từ một loạt các liên kết mà người dùng đã đi qua sẽ không giống như lật trang sách tuần tự. Người dùng phải luôn có khả năng quay trở lại trang chủ và các điểm chủ chốt trên Website dù họ đang ở bất kỳ đâu. Các liên kết cơ bản này nên có trên mọi trang Web của chúng ta, nó thường là các nút ẩn đồ họa với hai mục đích: tạo các mối liên kết cơ bản và giúp tạo một biểu tượng đồ họa thông báo cho người dùng biết họ vẫn còn đang ở trong Website của chúng ta.

- **Cho phép truy nhập trực tiếp**

Mục đích là cung cấp cho người dùng thông tin họ cần với ít bước nhất và với thời gian ngắn nhất. Điều này có nghĩa là chúng ta cần thiết kế cấu trúc thông tin một cách hiệu quả, giảm tối đa các bước qua hệ thống menu. Các nghiên cứu về giao diện chỉ ra rằng người dùng thích các menu có từ 5 đến 7 liên kết, và người dùng cũng ít thích màn hình dày đặc các lựa chọn hơn là nhiều trang với các menu đơn giản.

10.4.4. Tạo ngữ cảnh hoặc mất độc giả

Vấn đề chủ yếu của giao diện trong các Website là độc giả không ý thức được họ đang ở đâu trong tổ chức thông tin, không nắm được sơ đồ, cấu trúc của Website. Người dùng cần cảm nhận ngữ cảnh, về vị trí của họ trong tổ chức thông tin. Trong các tài liệu trên giấy, cảm giác "ta đang ở đâu" là sự phối hợp các cách xử lý về biên tập, đồ họa có được từ thiết kế sách, cách tổ chức văn bản và cảm giác vật lý của cuốn sách. Các tài liệu điện tử không cung cấp một ám chỉ vật lý nào cho việc truy nhập thông tin. Khi chúng ta thấy một liên kết Web trên một trang, chúng ta ít có cảm nhận chúng ta sẽ được dẫn đi đâu, có bao nhiêu thông tin ở đó, và chính xác thông tin đó quan hệ thế nào đến trang hiện tại. Một số không nhỏ các trang Web không vừa khớp với màn hình 14-15 inch, và do đó luôn có một phần của trang mà người dùng không thể nhìn thấy.

Các trang Web cần cho người dùng cảm nhận rõ ràng ngữ cảnh và tổ chức thông tin, vì chỉ có một phần nhỏ của Website (ít hơn một trang) được hiển thị vào một thời điểm. Nếu là một người thiết kế Web, chúng ta cần chuẩn bị để cung cấp cho độc giả các khả năng này.

10.4.5. Dài thông và ảnh hưởng

Độc giả không chịu đựng thời gian trễ dài. Các nghiên cứu nhân tố con người cho

thấy đối với đa số công việc tính toán, ngưỡng của sự mất tác dụng là khoảng 10 giây. Các thiết kế trang Web mà không thích hợp với tốc độ truy nhập mạng của người dùng sẽ làm cho họ thất vọng. Nếu người dùng chỉ là một người duyệt Web bình thường sử dụng modem tốc độ 28,8 hoặc 56 kbps qua đường điện thoại, thật đại dột khi đặt một ảnh lớn lên trang Web vì người dùng sẽ không đủ kiên nhẫn chờ đến khi ảnh được nạp xuống. Thế nhưng, nếu chúng ta xây dựng một Website cho nội bộ trường học, doanh nghiệp, nơi mà mọi người truy nhập Web server với tốc độ mạng LAN hay cao hơn nữa, thì chúng ta lại nên sử dụng nhiều ảnh và multimedia.

10.4.6. Đơn giản và nhất quán

Độc giả sẽ không ấn tượng với sự phức tạp không lý do, đặc biệt các người dùng phụ thuộc vào Website của chúng ta về thời gian hoặc thông tin chính xác, liên quan đến công việc. Các biểu tượng nên đơn giản, quen thuộc và dễ hiểu với độc giả, ví dụ nếu chúng ta muốn một biểu tượng cho việc lưu trữ thông tin, nên chọn quyển sách hay thư viện, chứ đừng chọn tàu vũ trụ hay võ tuyến truyền hình.

Adobe (do Studio Archetype thiết kế) là một mô hình xuất sắc cho thiết kế Web site. Các trang sử dụng đồ họa một cách rộng rãi với các trợ giúp việc duyệt, áp dụng nhất quán ở tất cả các trang trên Website. Một khi chúng ta biết các liên kết cơ bản nằm trên đầu trang, ranh giới trở thành vô hình và việc định vị sẽ dễ dàng hơn.

Để đạt được chức năng tối đa và tính rõ ràng, thiết kế trang của chúng ta nên được xây dựng trên các đơn vị mẫu nhất quán, tất cả đều sử dụng chung một kiểu đồ họa, phong cách biên tập và mô hình tổ chức. Mục đích là tính đồng nhất, có thể đoán trước, để người dùng cảm thấy thoải mái khi khai thác Website của chúng ta, và chắc chắn họ biết phải tìm cái họ cần. Các biểu tượng đồ họa trên trang Web cung cấp các tín hiệu trực quan về sự tiếp nối của thông tin. Biểu tượng đồ họa trên tất cả đầu trang của Adobe site tạo nên một giao diện người dùng nhất quán, và cũng là nét nhận dạng duy nhất cho site của Adobe.

Ngay cả khi các trang của chúng ta không sử dụng đồ họa, sử dụng nhất quán các tiêu đề, các chân trang và các liên kết đến trang chủ, các trang liên quan sẽ tăng cường cảm giác của độc giả là họ đang trong khung cảnh Web site của chúng ta.

Để tạo hiệu quả cho một hệ thống "không đứt đoạn" (*seamless*), chúng ta nên xem xét việc đưa các thông tin quan trọng vào Website của chúng ta, gộp nó vào hệ thống hơn là tạo liên kết đưa người dùng ra khỏi site của chúng ta.

10.4.7. Tính ổn định thiết kế

Nếu chúng ta mong muốn thuyết phục độc giả của chúng ta rằng cái mà chúng ta cung cấp là chính xác, đáng tin cậy, chúng ta cũng phải thiết kế Website của chúng ta cẩn thận, giống như chúng ta tạo các mối liên hệ với các doanh nghiệp khác, với cùng sự biên tập và các trình độ thiết kế cao. Một site trông luộm thuộm, với thiết kế trực quan nghèo nàn, trình độ biên tập kém sẽ không truyền được sự tin cậy cho các độc giả. Tính ổn định chức năng trong thiết kế Web có nghĩa là giữ các thành phần giao tiếp của Website làm việc

ổn định. Tính ổn định chức năng có hai thành phần - đặt các vật đúng chỗ ngay từ đầu khi thiết kế Website, và sau đó giữ chúng hoạt động nhịp nhàng trong suốt thời gian. Các Web site tốt tự nó đã có tác động qua lại, với nhiều liên kết đến các trang trong site đó, và có các liên kết đến các site khác. Trong khi thiết kế, chúng ta cần kiểm tra thông xuyên các liên kết để đảm bảo chúng còn tồn tại vì mọi cái trên Web thay đổi rất nhanh, cả trên Website của chúng ta và các Website khác. Chúng ta cần có lịch kiểm tra lại các liên kết và cả nội dung của nó có còn thích hợp không.

10.4.8. Phản hồi và đối thoại

Thông qua hình ảnh đồ họa, các nút nhấn, các liên kết đặt một nơi duy nhất, thiết kế Web của chúng ta nên đưa ra khả năng xác nhận vị trí, lựa chọn của độc giả. Phản hồi cũng có nghĩa là bước chuẩn bị cho việc trả lời, đáp ứng các đòi hỏi, góp ý của độc giả. Các Website thiết kế tốt luôn cung cấp mỗi liên hệ trực tiếp tới ban biên tập hoặc "Webmaster" phụ trách kỹ thuật của site. Lên kế hoạch đảm bảo quan hệ liên tục với các độc giả là quan trọng sống còn đối với sự thành công lâu dài của một doanh nghiệp.

10.4.9. Thiết kế cho các trình duyệt khác

Không phải mọi độc giả của chúng ta đều có thể được hưởng lợi từ các ảnh đồ họa chúng ta đưa ra trên trang Web, và rất nhiều độc giả không có chương trình duyệt đồ họa. Một trong những cái hay của Web và HTML là khả năng thay thế thông báo (nhãn ALT trong HTML) để các độc giả với trình duyệt không có khả năng đồ họa (lynx) vẫn hiểu được chức năng của hình ảnh trên trang Web. Sử dụng chương trình đặc biệt, các độc giả mù có thể nghe các thông báo thay thế mà chúng ta cho hỗ trợ cùng hình ảnh đồ họa, do đó không hoàn toàn mất hẳn nội dung của bức ảnh, phím đồ họa đi kèm trang Web.

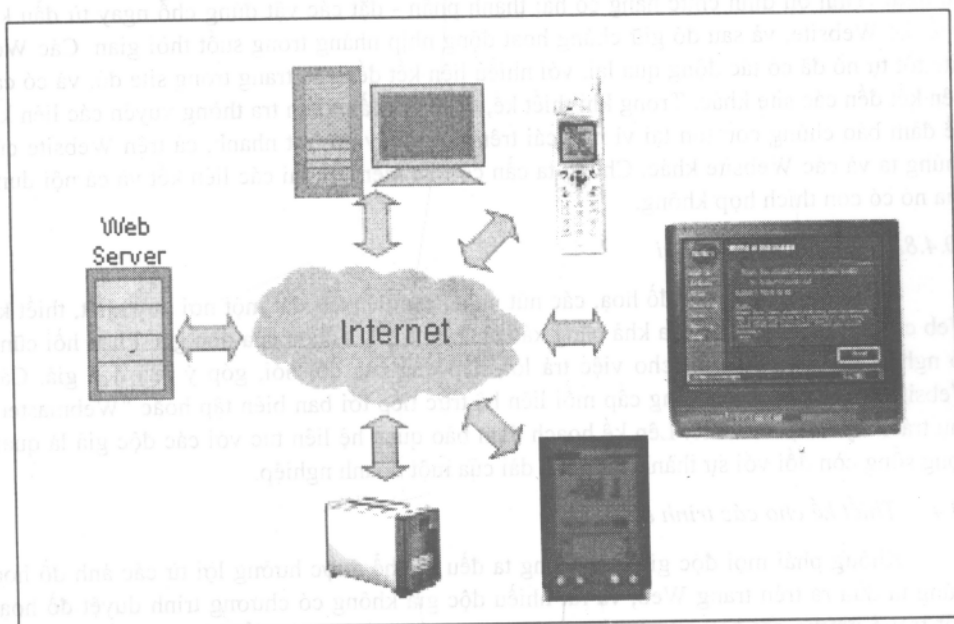
10.4.10. Chú ý cuối cùng

Hãy suy nghĩ chín chắn trước khi đưa vào website một trang đặc biệt với nhiều hình ảnh màu sắc hay hoạt hình sống động. Nếu công việc của ta có liên quan tới thiết kế đồ họa, các hình ảnh động hoặc một cái gì đó đòi hỏi phải thể hiện các kỹ năng kỹ xảo về trực quan thì một trang như vậy sẽ là thích hợp. Còn đối với các trang mang tính chất kinh doanh hay học tập thì phần lớn sẽ không như vậy. Hãy luôn nhớ rằng hầu hết khách hàng không vào những trang loại này để vui chơi giải trí, họ tới vì các thông tin, vì nội dung của trang Web và việc đưa vào nhiều những hình thức màu mè, không cần thiết không chỉ làm chậm tốc độ mà còn khiến họ bực mình.

10.5. Một số xu hướng phát triển của tương tác người máy trong WEB

10.5.1. Tính độc lập thiết bị (Device Independence)

Mọi thiết bị được đảm bảo truy cập vào Web một cách trong suốt. Các dịch vụ Web có thể truy cập được từ mọi thiết bị như máy tính để bàn, máy tính trong ôtô, ti vi, máy ảnh số, điện thoại di động. W3C định hướng sự phát triển để tránh sự không tương thích và để đạt được sự độc lập thiết bị trong truy cập Web (hình 10.10)



Hình 10.10. Tính đa dạng của thiết bị.

10.5.2. Tương tác đa phương thức (Multimodal Interaction)

Mở rộng giao diện người dùng Web cho phép nhiều phương thức giao tiếp khác nhau, cung cấp cho người dùng sự lựa chọn giữa việc sử dụng giọng nói, hoặc sử dụng vùng phím số, bàn phím, chuột, stylus hoặc các thiết bị vào khác. Đối với đầu ra, người dùng có thể nghe âm thanh, và để xem thông tin trên các thiết bị hiển thị đồ họa.

10.5.3. Đồng bộ đa phương tiện (Synchronized Multimedia)

Tập trung vào việc thiết kế một ngôn ngữ cho kết hợp âm thanh, video, văn bản và đồ họa trong thời gian thực để tạo ra một sự thể hiện ấn tượng. Ngôn ngữ, SMUL (Synchronized Multimedia Untegration Language) được viết như một ứng dụng XML. Nó cho phép tác giả chỉ định cái gì cần được hiển thị và vào lúc nào. SMUL 1.0 đang được W3C đề xuất, và phiên bản thứ hai đang được phát triển.

10.5.4. Trình duyệt bằng giọng nói (Voice Browser)

Cho phép người dùng tương tác với các trang Web thông qua tiếng nói, và nghe âm thanh được ghi lại, âm nhạc, và tổng hợp giọng nói. Điều đó cho phép sử dụng điện thoại truy cập các dịch vụ Web. Đồng thời, nó cho phép người sử dụng truy cập Web trong khi mất và tay được sử dụng cho các việc khác.

10.5.5. Người mù qua mạng

Đây là khía cạnh mới bắt đầu được nghiên cứu: máy tính lưu sẵn một số mẫu mùi cơ bản, khi người dùng tra cứu Web có hình ảnh tương ứng với một số mùi nào đó trong máy tính, ta sẽ ngửi được mùi đó.

10.6. Kết luận

Do bản chất không tuyến tính của siêu văn bản trong giao tiếp cùng với việc sử dụng các phương tiện khác nhau tạo nên một tổ hợp mạnh trong giáo dục, trong văn bản hoá và trong quảng cáo. Hơn nữa, Web đã tạo nên một công cụ đại chúng như một vật liệu toàn cầu với một cái giá thấp. Tuy nhiên, cũng do cấu trúc phi tuyến của Web, người dùng có thể bị nhầm lẫn và mất phương hướng nếu cấu trúc của Web không rõ ràng. Những vấn đề này sẽ trở nên nghiêm trọng trong các ứng dụng dựa vào Web, nơi các liên kết có thể vượt khỏi mạng cục bộ hay vượt phạm vi quốc gia. Hơn nữa các trang Web thường được phát triển rất nhanh và thiếu kế hoạch.

Việc sử dụng hình ảnh, hoạt hình cũng có thể và nhất là trong một số ứng dụng đặc biệt. Video cũng có thể sử dụng. Tuy nhiên cần chú ý vấn đề lưu trữ, tốc độ và băng thông.

11

THIẾT KẾ TRỢ GIÚP

Nội dung

- 11.1. Tổng quan về trợ giúp
- 11.2. Các yêu cầu về hỗ trợ người dùng
 - 11.2.1. Tính sẵn dùng
 - 11.2.2. Tính chính xác và đầy đủ
 - 11.2.3. Tính nhất quán
 - 11.2.4. Tính vững chắc
 - 11.2.5. Tính mềm dẻo
 - 11.2.6. Tính không lạc hậu?
- 11.3. Các tiếp cận để hỗ trợ người dùng
 - 11.3.1. Lệnh trợ giúp
 - 11.3.2. Dấu nhắc lệnh
 - 11.3.3. Trợ giúp ngữ cảnh
 - 11.3.4. Huấn luyện trực tuyến
 - 11.3.5. Tài liệu trực tuyến
- 11.4. Hệ thống trợ giúp thích nghi
 - 11.4.1. Biểu diễn tri thức: mô hình người dùng
 - 11.4.2. Biểu diễn tri thức: mô hình hoá nhiệm vụ và lĩnh vực
 - 11.4.3. Biểu diễn tri thức: mô hình hoá chiến lược tư vấn
 - 11.4.4. Các kỹ thuật biểu diễn tri thức
- 11.5. Thiết kế hệ thống trợ giúp người dùng
 - 11.5.1. Mô thức trình diễn
 - 11.5.2. Mô thức cài đặt
- 11.6. Ví dụ thiết hệ trợ giúp dạng HTML

11.1. Tổng quan về trợ giúp

Một ứng dụng dù có hoàn hảo tới mức nào đi nữa, khi đưa tới người sử dụng cũng không đảm bảo được rằng họ sẽ sử dụng thành thạo nó. Vì người sử dụng không phải là người thiết kế ra phần mềm nên họ không thể nào nắm bắt được hết các chức năng, thậm chí họ còn không biết sử dụng phần mềm đó như thế nào. Để hỗ trợ người dùng sử dụng tối đa lợi ích của phần mềm, hầu hết các hệ thống phần mềm ngày nay đều cung cấp thêm hệ thống trợ giúp Help. Hệ thống trợ giúp đã trở thành một thành phần quan trọng trong sản phẩm phần mềm. Một hệ thống trợ giúp tốt sẽ là cầu nối giữa sự phức tạp của phần mềm và sự đơn giản trong sử dụng của khách hàng.

Người dùng sẽ yêu cầu nhiều loại trợ giúp, nó phụ thuộc vào nhiều nhân tố như sự hiểu biết của người dùng về hệ thống, công việc mà họ đang cố gắng làm,... Có bốn loại trợ giúp chính mà người sử dụng yêu cầu:

- 1 Tham khảo nhanh - *Quick reference*.
- 2 Trợ giúp cho từng công việc - *Task specific help*.
- 3 Giải thích đầy đủ - *Full explanation*.
- 4 Hướng dẫn - *Tutorial*.

Quick reference được sử dụng chủ yếu để gợi nhớ cho người sử dụng về chi tiết các công cụ mà anh ta đã biết rõ và đã sử dụng trước đây. Ví dụ nó có thể được sử dụng để tìm các lựa chọn cho các câu lệnh hoặc gợi nhớ cho người dùng cú pháp của câu lệnh.

Task-specific help được yêu cầu khi người sử dụng bắt gặp một vấn đề trong khi thực hiện các công việc riêng hoặc khi anh ta không biết phải làm thế nào để sử dụng công cụ nhằm giải quyết vấn đề của mình. Sự giúp đỡ được đề nghị có quan hệ trực tiếp tới những gì sẽ được thực hiện.

Người sử dụng có nhiều kinh nghiệm hay người tò mò thường yêu cầu *lời giải thích đầy đủ* về công cụ hay câu lệnh cho phép anh ta có thể hiểu đầy đủ hơn. Lời giải thích này hầu như chắc chắn có cả những thông tin mà người dùng không cần đến trong lúc đó.

Loại hỗ trợ thứ tư mà người dùng yêu cầu là trợ giảng (*Tutorial*). Loại này hướng dẫn người mới dùng về phần mềm và cung cấp chỉ dẫn từng bước một hướng dẫn sử dụng phần mềm.

Mỗi loại hỗ trợ người dùng là sự bổ sung kiến thức và được yêu cầu ở những thời điểm khác nhau trong những kinh nghiệm khác nhau của người dùng đối với hệ thống. Trong các loại hỗ trợ mà người dùng yêu cầu sẽ có nhiều thông tin mà người dùng cần như các định nghĩa, các ví dụ, các lỗi và phục hồi lỗi, các lựa chọn câu lệnh... Một trong những số đó có thể được cung cấp cùng thiết kế giao diện, số còn lại được bao trong phần trợ giúp hệ thống. Các loại trợ giúp khác nhau được yêu cầu gồm cả những điều cần thiết nhằm cung cấp các loại khác nhau của hệ thống giúp đỡ. Thường có sự khác biệt giữa hệ thống

giúp đỡ và hệ thống các tư liệu. Hệ thống trợ giúp là hệ thống hướng sự kiện và riêng biệt, trong khi đó hệ thống các tư liệu là hướng hệ thống và chung chung.

11.2. Các yêu cầu về hỗ trợ người dùng

Một hệ trợ giúp sẽ như thế nào? Đây là một câu hỏi khó trả lời song chúng ta sẽ đi qua vài nét đặc trưng chính của một hệ trợ giúp. Không phải mọi hệ trợ giúp đều phải có đầy đủ các đặc điểm đó nhưng chúng phải khá đầy đủ để ta có thể kiểm tra các công cụ hỗ trợ mà ta thiết kế. Nếu một hệ trợ giúp không đủ các tính năng đó thì ta phải thiết kế lại. Sau đây là các đặc trưng chính mà một hệ trợ giúp phải có.

11.2.1. Tính sẵn dùng

Người sử dụng cần sự trợ giúp bất cứ lúc nào trong thời gian tương tác với hệ thống. Thông thường, anh ta không phải đóng ứng dụng để mở help. Một hệ thống trợ giúp phải luôn luôn sẵn dùng để khi có yêu cầu có thể lấy ra để sử dụng.

11.2.2. Tính chính xác và đầy đủ

Rất rõ ràng rằng hệ trợ giúp được cung cấp phải chính xác và đầy đủ. Có như vậy người sử dụng mới có thể hiểu đúng đắn và đầy đủ về hệ thống. Một ứng dụng thường xuyên được cập nhật, qua các phiên bản khác nhau, đó chưa phải là vấn đề. Tuy nhiên, nếu hệ trợ giúp được cải tiến không phù hợp với các phiên bản thì sẽ gây khó khăn cho người sử dụng. Một ứng dụng sẽ chẳng có ý nghĩa gì nếu như người sử dụng không biết nó dùng để làm gì và nếu có biết thì phải làm như thế nào. Nếu đã có trợ giúp nhưng nó không chính xác thì điều đó còn nguy hiểm hơn, bởi vì khi đó người sử dụng sẽ sử dụng phần mềm sai mục đích. Một hệ thống sẽ được phản ánh tốt nếu như hệ trợ giúp phủ toàn bộ ứng dụng-tức là tính đầy đủ. Tính đầy đủ cũng rất quan trọng khi nhà cung cấp hệ trợ giúp muốn có hiệu quả, những người thiết kế không thể nào dự đoán được phần nào của hệ thống người dùng sẽ yêu cầu trợ giúp cho nên phải thiết kế hệ trợ giúp sao cho đầy đủ. Việc không tìm thấy help cho một chủ đề sẽ làm người sử dụng thất vọng.

11.2.3. Tính nhất quán

Như ta đã biết, người dùng yêu cầu những loại trợ giúp khác nhau cho các mục đích khác nhau. Điều này nói lên rằng một hệ trợ giúp cần phải được kết hợp chặt chẽ giữa các phần. Phần trợ giúp trong mỗi phần phải nhất quán với các phần khác. Ví dụ như trợ giúp trực tuyến (online-help) phải nhất quán với các tài liệu văn bản. Cần có sự nhất quán trong các chỉ mục, các thuật ngữ, và các kiểu biểu diễn. Nó còn là kết quả của ứng dụng có hỗ trợ người dùng, nó phải nhất quán giữa các hệ thống. Một hệ trợ giúp sẽ không phải là mạnh nếu như một câu lệnh được mô tả thế này ở chỗ này nhưng lại là thế kia ở chỗ khác - tức là không nhất quán.

11.2.4. Tính vững chắc

Hệ thống trợ giúp thường được sử dụng bởi những người khó khăn trong sử dụng, có

Chương 11: Thiết kế trợ giúp

lẽ là do hệ thống thường tạo ra các hành vi bất thường hay các lỗi. Điều đó rất quan trọng để sau này hệ thống trợ giúp phải có khả năng chịu lỗi, cả sửa lỗi trình bày lẫn các hành vi có thể dự đoán được. Do đó người dùng có thể tin cậy vào khả năng có sự trợ giúp khi có yêu cầu. Trong thực tế, tính chịu lỗi, vì lý do đó, thậm chí còn quan trọng cho hệ trợ giúp hơn cả các phần khác.

11.2.5. Tính mềm dẻo

Rất nhiều hệ trợ giúp khá cứng nhắc, chúng có những thông điệp trợ giúp tương tự nhau bất chấp sự thành thạo của một số người đang tìm kiếm trợ giúp hay tình huống mà họ đang làm việc. Một hệ trợ giúp linh hoạt sẽ cho phép người sử dụng tìm được những thông tin phù hợp. Điều này được thể hiện trong quá trình thiết kế một hệ thống trợ giúp, và xuyên suốt trong trợ giúp theo ngữ cảnh.

11.2.6. Tính không cản trở

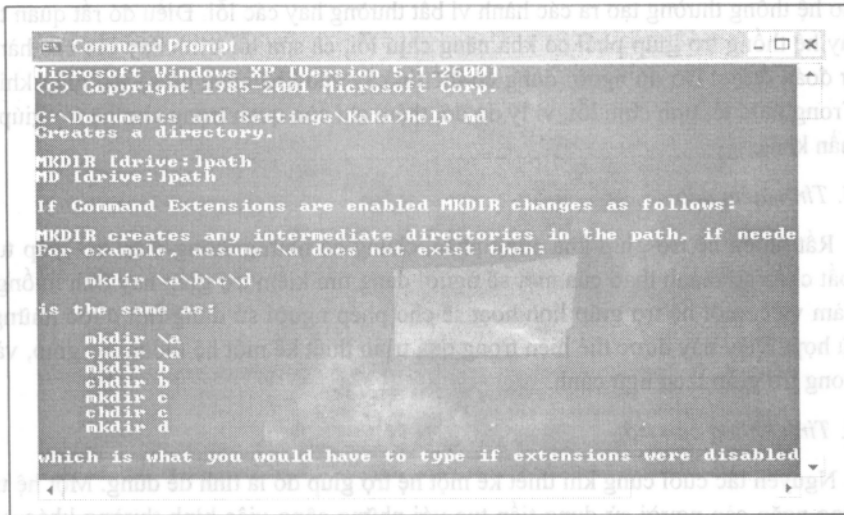
Nguyên tắc cuối cùng khi thiết kế một hệ trợ giúp đó là tính dễ dùng. Một hệ trợ giúp sẽ không ngăn cản người sử dụng tiếp tục với những công việc bình thường khác, và cũng không cản trở những ứng dụng của người dùng. Trong bất kỳ hệ thống giao diện không phải Windows sẽ làm ngất công việc của người sử dụng. Một giải pháp khả thi là chia đôi của sổ màn hình trình diễn. Lúc công việc khác kết thúc, một hệ thống trợ giúp có thể cung cấp sự trợ giúp tức thì, đúng hơn là khi có yêu cầu từ người sử dụng.

11.3. Các tiếp cận trong hỗ trợ người dùng

Các phương pháp tiếp cận khác nhau khi cung cấp đều có những sự riêng biệt cần thiết khác nhau. Điều này chúng ta sẽ thấy từ những gợi ý đơn giản cho đến những hướng dẫn đầy đủ cho hệ thống. Trong phần này ta sẽ tập trung vào những loại trợ giúp được cung cấp từ những hệ thống riêng biệt, được miêu tả chi tiết.

11.3.1. Lệnh trợ giúp

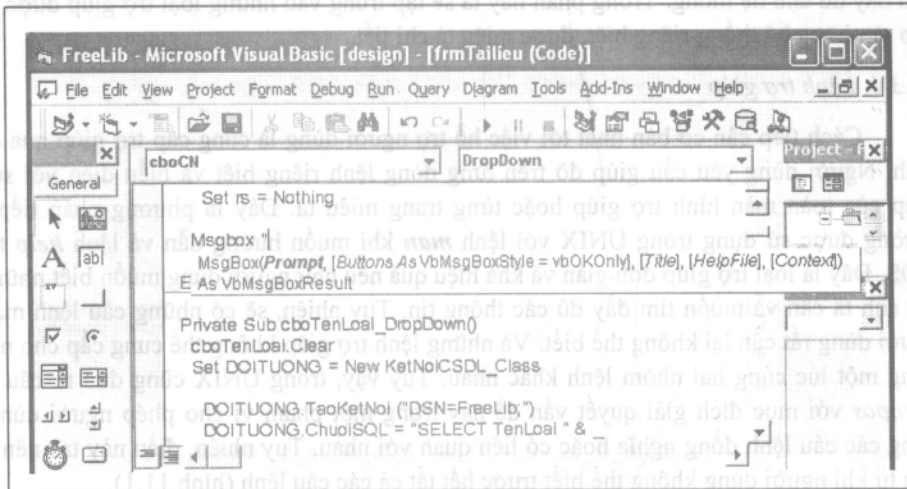
Cách tiếp cận cơ bản nhất tới việc hỗ trợ người dùng là cung cấp trợ giúp qua dòng lệnh. Người dùng yêu cầu giúp đỡ trên từng dòng lệnh riêng biệt và biểu diễn với sự trợ giúp của toàn màn hình trợ giúp hoặc từng trang miêu tả. Đây là phương pháp tiếp cận thường được sử dụng trong UNIX với lệnh *man* khi muốn hướng dẫn và lệnh *help* trong DOS. Đây là loại trợ giúp đơn giản và khá hiệu quả nếu như người dùng muốn biết những gì mà anh ta cần và muốn tìm đầy đủ các thông tin. Tuy nhiên, sẽ có những câu lệnh mà khi người dùng rất cần lại không thể biết. Và những lệnh trợ giúp không thể cung cấp cho người dùng một lúc cùng hai nhóm lệnh khác nhau. Tuy vậy, trong UNIX cũng đưa ra câu lệnh *apropos* với mục đích giải quyết vấn đề này trong một phạm vi cho phép người dùng sử dụng các câu lệnh đồng nghĩa hoặc có liên quan với nhau. Tuy nhiên, điều này trở nên giới hạn từ khi người dùng không thể biết trước hết tất cả các câu lệnh (hình 11.1).



Hình 11.1. Trợ giúp dòng lệnh trong DOS.

11.3.2. Dấu nhắc lệnh

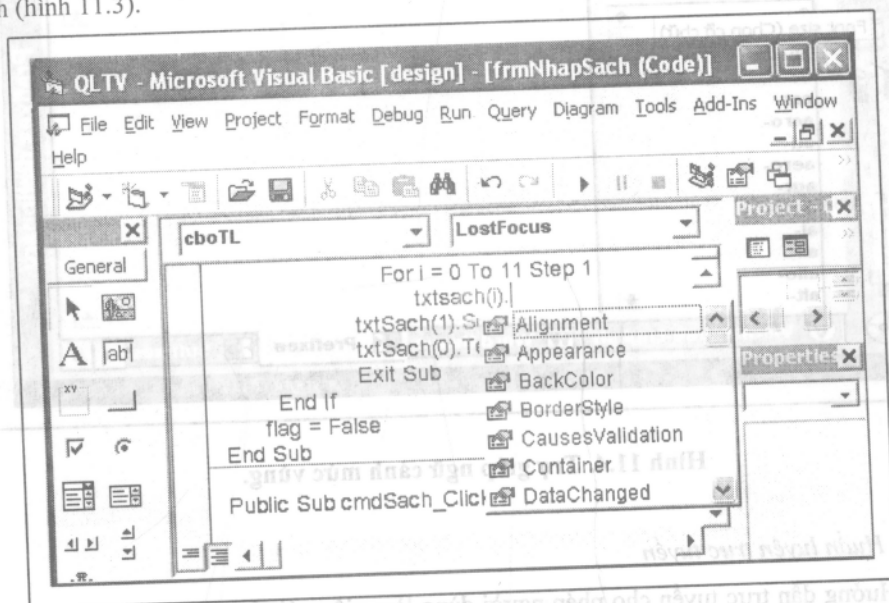
Dấu nhắc lệnh cung cấp sự trợ giúp khi người dùng bắt gặp một lỗi, thường ở trong dạng dấu nhắc sửa lỗi. Mỗi lời nhắc là khá hữu ích nếu như đó là lỗi đơn, như lỗi cú pháp. Hoặc là khi hướng dẫn cho người sử dụng cú pháp của câu lệnh, điều này được thể hiện khá rõ trong môi trường lập trình.



Hình 11.2. Một thí dụ về trợ giúp trong Visual Basic.

Chương 11: Thiết kế trợ giúp



Một dạng khác của việc nhắc lỗi, không có ý định đặc biệt để cung cấp nhưng hỗ trợ người sử dụng với mức độ hạn chế, được sử dụng là **Menu** và các biểu tượng có thể lựa chọn. Nó cung cấp sự hỗ trợ để người dùng có thể nhớ các câu lệnh có sẵn một cách dễ dàng. Tuy nhiên, cũng phải thừa nhận rằng người sử dụng cần một số các kiến thức về câu lệnh (hình 11.3).

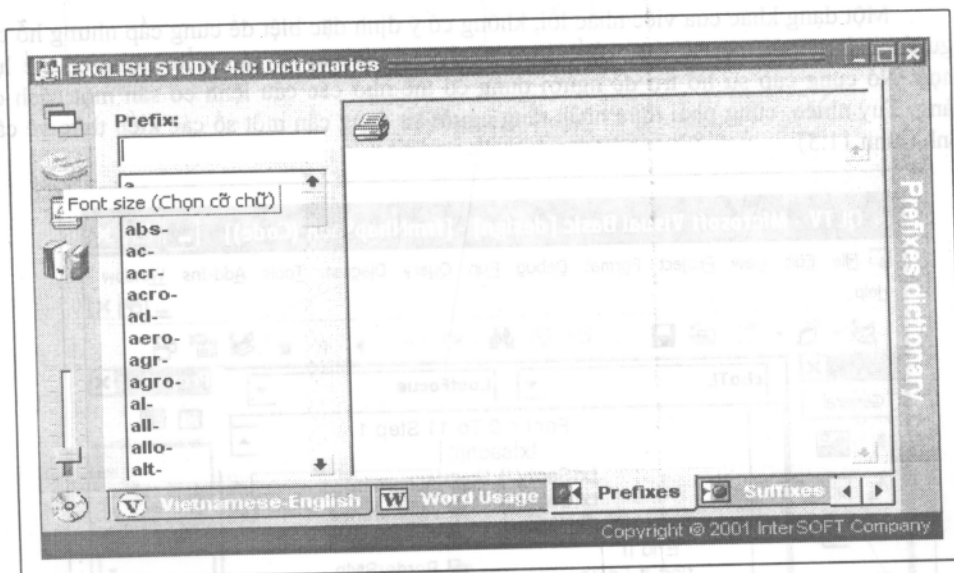


Hình 11.3. Trợ giúp kiểu menu của VB.

11.3.3. Trợ giúp ngữ cảnh

Một số hệ trợ giúp ở dạng ngữ cảnh. Phạm vi của các hệ trợ giúp này có kiến thức về người dùng với sự cung cấp các khóa hay chức năng được thông dịch tùy thuộc vào ngữ cảnh mà nó được gọi và sẽ thực hiện các trợ giúp phù hợp. Những hệ thống này không cần thiết phải phức tạp, song nó giúp người dùng không phải nhớ các câu lệnh. Có hai loại trợ giúp ngữ cảnh là trợ giúp mức window và trợ giúp mức vùng. Cách thứ nhất là dạng trợ giúp dựa trên các lựa chọn menu. Khi người sử dụng di chuyển con trỏ đến menu hay một lựa chọn nào đó, màn hình sẽ hiển thị một cửa sổ mới với thông tin về những chức năng hay những vấn đề mà người dùng có thể chưa biết (thông thường hiển thị ở dạng Tool Tip Text - hình 11.4).

Dạng thứ hai là trợ giúp ngữ cảnh kiểu Window. Tức là chương trình sẽ hiển thị các topic trợ giúp tương ứng khi người dùng ấn F1 hoặc lựa chọn chức năng Help hay click vào nút  hoặc nút  trên một cửa sổ, một hộp thoại, hay một vùng nào đó khi sử dụng các ứng dụng.

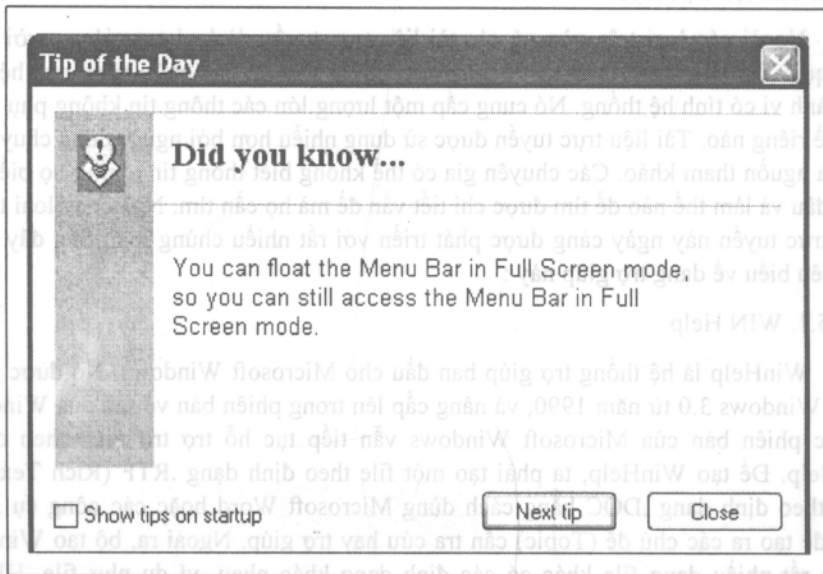


Hình 11.4. Trợ giúp ngữ cảnh mức vùng.

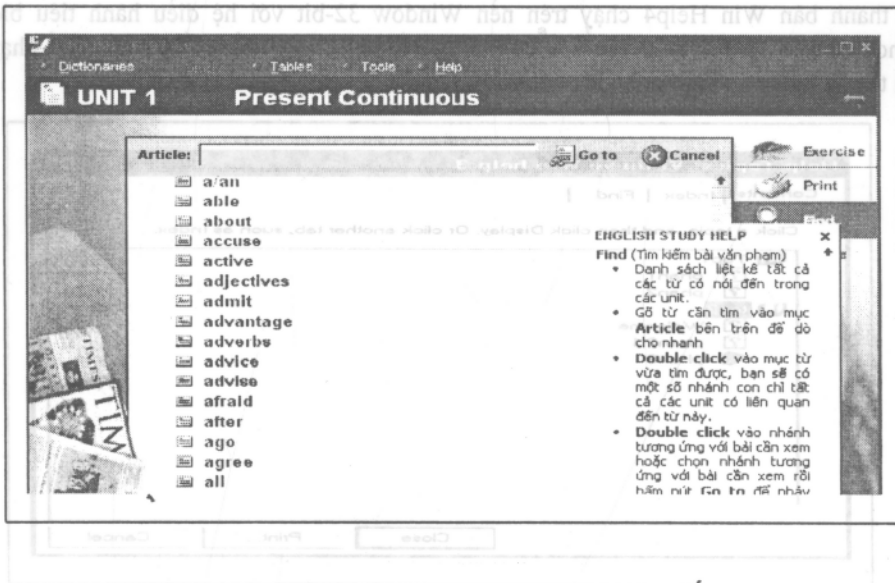
11.3.4. Huấn luyện trực tuyến

Hướng dẫn trực tuyến cho phép người dùng làm việc với những trợ giúp ngay trong ứng dụng. Người sử dụng có thể tiến hành tra cứu và lặp lại phần đó nếu thấy cần. Anh ta sẽ cảm nhận được công việc bằng việc thực hành với các ví dụ. Hầu hết các hướng dẫn trực tuyến không “thông minh”: chúng không biết gì về người dùng và kinh nghiệm trước đây của anh ta, cũng không biết yêu cầu cụ thể của người dùng, mà chỉ đưa ra những hướng dẫn chung chung về các chức năng hay công việc của chương trình. Hệ hướng dẫn thông minh, được sử dụng các kỹ thuật tương tự với hệ trợ giúp thích nghi, cố gắng đánh địa chỉ kết quả nhưng tách biệt với các ứng dụng lập trình hướng dẫn, không thực tế cho hầu hết các ứng dụng. Mặc dù vậy kiểu hướng dẫn trực tuyến cũng rất linh hoạt và không dễ bỏ qua. Có một số sẽ nhận dạng sai câu trả lời, đơn giản bởi vì nó không được định dạng như mong đợi. Chẳng hạn, trong một số chương trình sẽ xuất hiện một cửa sổ hướng dẫn ngay khi người sử dụng truy cập vào chương trình đó. Người sử dụng có thể bỏ qua điều này nếu họ thấy không thực sự cần thiết. Hình 11.5 minh họa về dạng trợ giúp này.

Ngoài ra, khi một hệ thống hay một chương trình luôn có các phiên bản mới hay các chức năng mới thì sự hướng dẫn trực tuyến sẽ giúp cho người sử dụng hiểu về rõ hơn về hệ thống hay chương trình đó. Người sử dụng sẽ tìm thấy sự giúp đỡ ngay lập tức nếu anh ta thấy cần thiết (hình 11.6).



Hình 11.5. Một dạng trợ giúp trực tuyến.



Hình 11.6. Một phiên bản của trợ giúp trực tuyến.

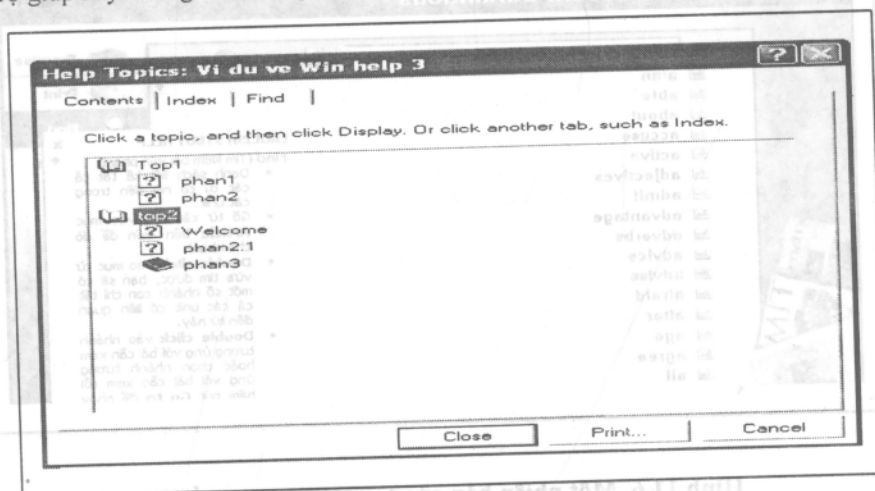
Chương 11: Thiết kế trợ giúp

11.3.5. Tài liệu trực tuyến

Ngoài các loại trên còn có các tài liệu trực tuyến, là loại trợ giúp người dùng khá hiệu quả. Tài liệu được thiết kế để cung cấp mô tả đầy đủ các chức năng của hệ thống và các hành vi có tính hệ thống. Nó cung cấp một lượng lớn các thông tin không phụ thuộc vào vấn đề riêng nào. Tài liệu trực tuyến được sử dụng nhiều hơn bởi người dùng chuyên nghiệp như là nguồn tham khảo. Các chuyên gia có thể không biết thông tin nhưng họ biết phải tìm nó ở đâu và làm thế nào để tìm được chi tiết vấn đề mà họ cần tìm. Ngày nay loại tài liệu trợ giúp trực tuyến này ngày càng được phát triển với rất nhiều chủng loại. Sau đây là một số loại tiêu biểu về dạng trợ giúp này :

11.3.5.1. WIN Help

WinHelp là hệ thống trợ giúp ban đầu cho Microsoft Windows. Nó được giới thiệu trong Windows 3.0 từ năm 1990, và nâng cấp lên trong phiên bản về sau của Windows. Tất cả các phiên bản của Microsoft Windows vẫn tiếp tục hỗ trợ trợ giúp theo định dạng WinHelp. Để tạo WinHelp, ta phải tạo một file theo định dạng .RTF (Rich Text Format) hoặc theo định dạng .DOC bằng cách dùng Microsoft Word hoặc các công cụ soạn thảo khác để tạo ra các chủ đề (Topic) cần tra cứu hay trợ giúp. Ngoài ra, bộ tạo Win help còn tạo ra rất nhiều dạng file khác có các định dạng khác nhau, ví dụ như file .HPJ (Help Project File) là file chứa toàn bộ nội dung được tạo ra khi thiết kế bộ trợ giúp, hoặc là loại file .CNT (Help Contents File) chứa các đề mục hay topic trong bộ trợ giúp; còn file .HLP là file hiển thị cho người sử dụng tra cứu. Bản Win Help đầu tiên là bản Win Help3 chạy trên nền Window 16-bit mà điển hình là thế hệ hệ điều hành Window 3.x, sau này nâng cấp lên thành bản Win Help4 chạy trên nền Window 32-bit với hệ điều hành tiêu biểu là Window 95, mặc dù vậy 2 bản này có những đặc tính tương đối giống nhau. Để chạy hai loại trợ giúp này không bắt buộc phải kết nối vào một mạng nào đó.



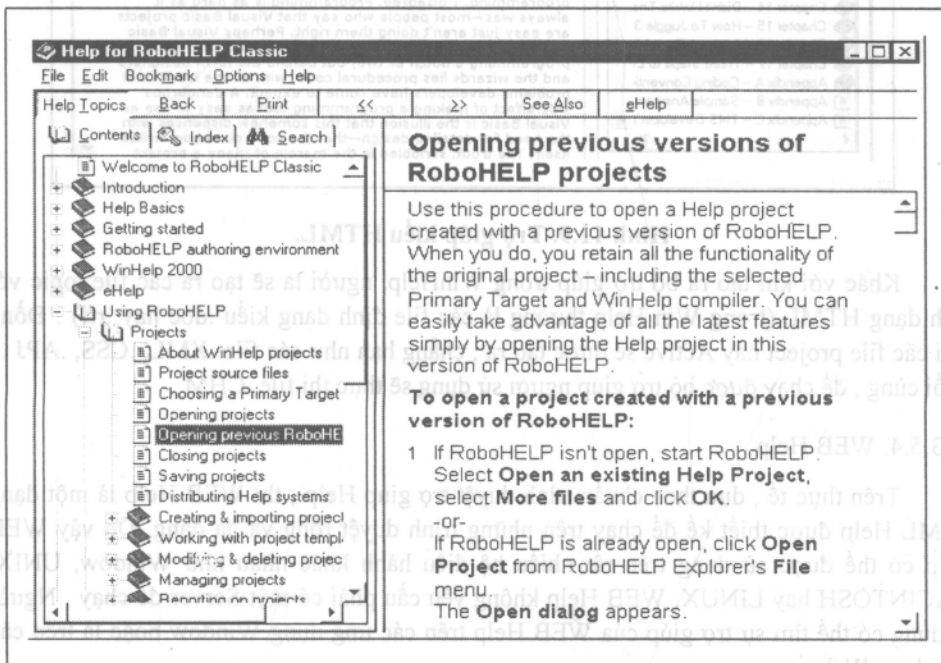
Hình 11.7 Trợ giúp của WINHELP

Chương 11: Thiết kế trợ giúp

Hai loại WinHelp này có hai cửa sổ riêng rẽ, một để hiển thị các topic, các chủ đề cần tra cứu, và một để hiển thị nội dung của topic đó. Người sử dụng muốn tra cứu một vấn đề gì thì tìm trước chủ đề ở cửa sổ thứ nhất sau đó nhấn nút Display thì cửa sổ thứ hai sẽ hiện chi tiết nội dung của chủ đề đó. Người sử dụng có thể luân chuyển qua lại giữa hai cửa sổ một cách dễ dàng. Sau khi người sử dụng chọn chủ đề ở cửa sổ này rồi nhấn nút Display thì sẽ bật sang cửa sổ thứ hai, hiển thị nội dung của chủ đề đó.

11.3.5.2. WIN Help 2000

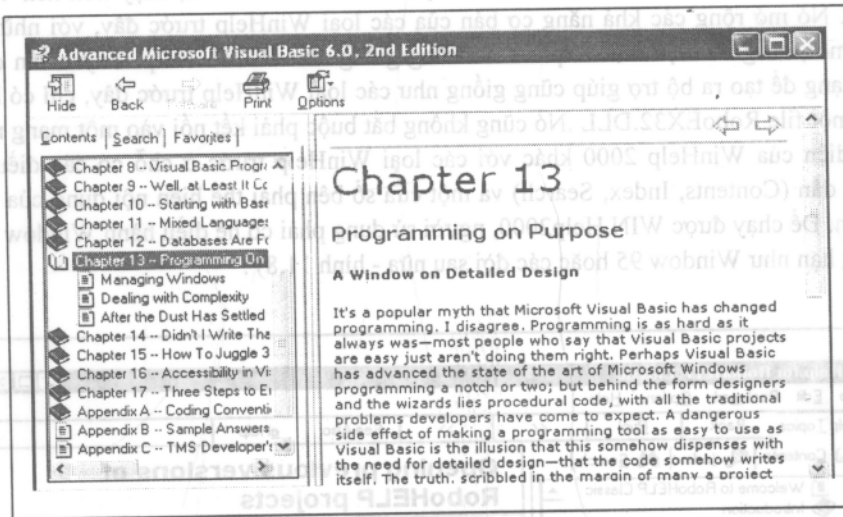
WinHelp 2000 được nghiên cứu và phát triển từ năm 1998, chạy trên nền Window 32-bit. Nó mở rộng các khả năng cơ bản của các loại WinHelp trước đây, với những tính năng mới, có giao diện như Explore và trông giống như HTML Help. Tuy nhiên các file định dạng để tạo ra bộ trợ giúp cũng giống như các loại WinHelp trước đây, chỉ có khác là thêm một file RoboEX32.DLL. Nó cũng không bắt buộc phải kết nối vào một mạng nào đó. Giao diện của WinHelp 2000 khác với các loại WinHelp trước ở chỗ có các điều khiển hướng dẫn (Contents, Index, Search) và một cửa sổ bên phải thể hiện nội dung của chủ đề cần tìm. Để chạy được WIN Help2000, người sử dụng phải có hệ điều hành Window 32-bit (chẳng hạn như Window 95 hoặc các đời sau nữa - hình 11.8).



Hình 11.8. Winhelp 2000.

11.3.5.3. HTML Help

Thể loại HTML Help được ra đời vào năm 1997 nhằm nâng cấp những chức năng mà các loại Win Help3 và Win Help4 trước đây chưa có. Nó cũng chạy trên nền Window 32-bit. HTML Help sẽ tích hợp 2 cửa sổ tra cứu vào làm một, do vậy giao diện sử dụng sẽ rất thân thiện với người dùng. HTML Help được sử dụng rộng rãi trong rất nhiều ứng dụng trên Window . Tuy nhiên để chạy được trên môi trường Window thì người sử dụng phải cài trình duyệt Internet Explorer 4.0 hoặc cao hơn nữa. Sau này Win Help 2000 được tạo ra cũng dựa theo một phần của HTML trợ giúp (hình 11.9).



Hình 11.9. Trợ giúp kiểu HTML.

Khác với khi tạo ra bộ trợ giúp trong WinHelp, người ta sẽ tạo ra các file topic với định dạng HTML (trong Win Help thường là các file định dạng kiểu .doc hay .rtf) . Đồng thời các file project hay Active sẽ được tạo ra , chẳng hạn như các file .XML, .CSS, .APJ ... Cuối cùng , để chạy được bộ trợ giúp người sử dụng sẽ thực thi file .CHM .

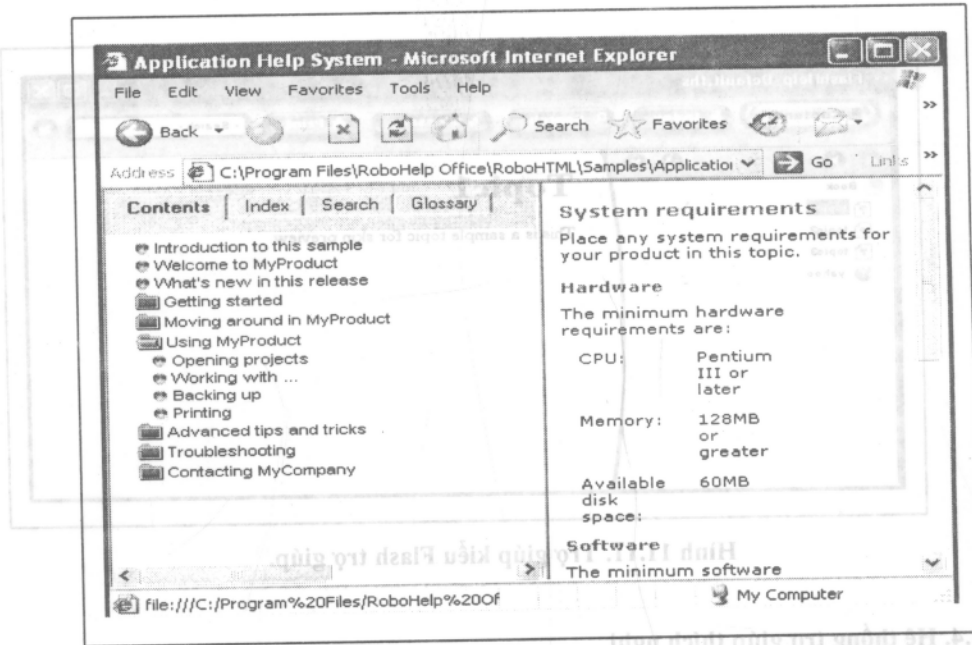
11.3.5.4. WEB Help

Trên thực tế , dựa theo chuẩn trình duyệt trợ giúp Help , thì WEB Help là một dạng HTML Help được thiết kế để chạy trên những trình duyệt rộng và đa dạng. Do vậy WEB Help có thể được sử dụng trên rất nhiều hệ điều hành khác nhau như Window, UNIX, MACINTOSH hay LINUX. WEB Help không yêu cầu phải có một Server để chạy . Người sử dụng có thể tìm sự trợ giúp của WEB Help trên các ứng dụng Window hoặc là trên các ứng dụng Web .

WEB help sẽ tạo ra các file định dạng HTML, XML, ... và các file imagine, chúng sẽ được phân bổ trên các thư mục của người sử dụng hoặc trên một server. WEB Help có thể

Chương 11: Thiết kế trợ giúp

chạy mà không cần sử dụng một phần mềm nào trên server. Để chạy được WEB Help thì phải có một bộ trình duyệt, chẳng hạn như Internet Explorer hay Netscape.



Hình 11.10. Trợ giúp kiểu WEB Help.

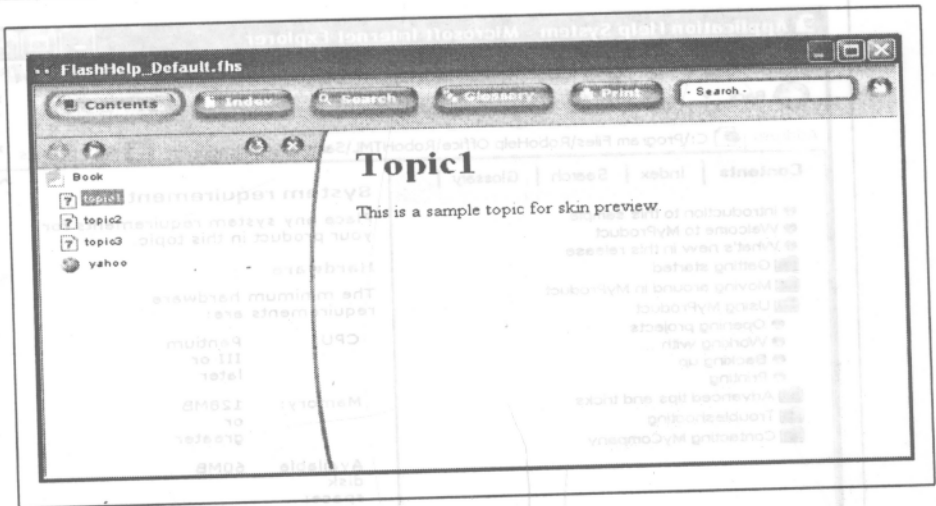
Ngoài ra còn có một dạng WEB Help nữa đó là dạng WEB Help Pro. Đây là một dạng Help server-based được thiết kế để hiển thị trong tất cả các trình duyệt lớn cũng như các nền đa dạng khác nhau như là WEB Help. Tuy nhiên các thành phần của WEB Help Pro có thêm các tính năng hết sức mạnh mẽ. Chẳng hạn như nó có thể cung cấp khả năng update các thông tin trợ giúp sau khi ứng dụng đã thoát; nó là một loại ngôn ngữ tìm kiếm tự nhiên mạnh. Người dùng phải cài đặt Microsoft IIS nếu muốn chạy WEB Help Pro.

11.3.5.5. Flash Help

Flash Help là thế hệ nâng cấp từ WEB Help, đây là một cuộc cách mạng về việc sử dụng Help trên các ứng dụng desktop (Window, Mac, Linux) và các ứng dụng trên Web, bao gồm cả .NET. Flash Help cung cấp những trình duyệt tìm kiếm hay các platform hết sức đặc sắc hơn hẳn các dạng Help đã từng có từ trước đây. Với các tính năng đặc biệt sau:

- Tìm kiếm nhanh, hiệu quả khi phải qua firewalls và băng thông kết nối thấp
- Tìm theo yêu cầu của người sử dụng
- Giáp diện Help động, có tích hợp các Flash sinh động bao gồm cả âm thanh

Các file mà Flash Help tạo ra là các file có định dạng HTML, XML và SWF. Các file này được lưu trữ trên các thư mục ở desktop hoặc trên Web server. Người sử dụng muốn chạy Flash Help phải có một trình duyệt (Internet Explorer) và Flash Player 6.0 trở lên.



Hình 11.11. Trợ giúp kiểu Flash trợ giúp.

11.4. Hệ thống trợ giúp thích nghi

Trong mọi hệ thống máy tính phức tạp hay lớn, người dùng sẽ quen với một hệ thống con của những tính năng sẵn có, và bày tỏ mối quan tâm với một số ứng dụng và không có kinh nghiệm với một số khác, cũng có thể ở mức độ không biết chúng có tồn tại không. Hơn nữa, những người dùng khác nhau sẽ có những nhu cầu khác nhau và ở mức độ khác nhau trong quá trình sử dụng hệ thống. Những hệ trợ giúp thích nghi cố gắng xác định những vấn đề này bằng cách thích ứng từng trợ giúp cho từng người dùng.

Hệ thống thích ứng là một trường hợp đặc biệt của một lớp tổng quát của những hệ thống tương tác, được biết đến như những hệ chuyên gia theo từng lĩnh vực, hệ thống giảng dạy thông minh và giao diện thích ứng tổng quát. Nhưng ở đây, chúng ta chỉ đề cập đến hệ thống trợ giúp thích ứng.

Hệ thống trợ giúp thích ứng vận hành vận hành bằng cách quan sát các hoạt động của người dùng và xây dựng nên một mô hình của mình. Đó có thể là một mô hình về kinh nghiệm, sở thích của người dùng hoặc có thể là mô hình với tổ hợp của một số đặc điểm hoặc tất cả. Sử dụng những tri thức này, cùng với tri thức về lĩnh vực người dùng đang làm, hệ thống trợ giúp thích ứng với tác vụ hiện thời của người dùng và phù với kinh nghiệm của họ. Tuy nhiên, trong thực tế nó không đơn giản. Thứ nhất, những yêu cầu tri thức cho những hệ thống như vậy là rất lớn và dữ liệu trên sự tương tác là rất khó thông dịch. Thứ hai, có

Chương II: Thiết kế trợ giúp

vấn đề đối với sự điều khiển và khởi tạo trong tương tác, vậy hệ thống trợ giúp có nên đảm nhận vai trò chủ động, tước bỏ một số quyền điều khiển khỏi người dùng, và những tính năng thích ứng này có làm bối rối người dùng nếu anh ta thấy nó thật là xa vời thực tế. Thứ ba, chính xác là những gì nên được coi là thích ứng để đưa ra, và kết quả của sự thích ứng này là gì. Và cuối cùng, phạm vi của sự thích ứng và mô hình hoá là gì: nó có đi quá xa khỏi mức độ ứng dụng, và nó có giải quyết được vấn đề về sự đa dạng trong việc thích ứng với mỗi người dùng trong suốt một hệ thống? Một số vấn đề này vẫn còn đang nghiên cứu. Dưới đây chúng ta sẽ xem xét một số vấn đề phát triển và giải pháp trong yêu cầu trí thức.

11.4.1. Biểu diễn tri thức: mô hình người dùng

Mọi hệ thống tương tác được xây dựng để kết hợp một số mô hình của người sử dụng mà nó hướng tới. Trong nhiều hệ thống, mô hình này là cách nhìn của người thiết kế với người dùng và được che đi trong thiết kế. Người thiết kế luôn có trong đầu một người dùng điển hình và xây dựng giao diện theo cách đó. Tuy nhiên, người ta cho rằng tất cả người dùng đều như nhau và đều có chung yêu cầu.

Những hệ thống khác cho phép người dùng cung cấp một mô hình của chính anh ta theo đó hệ thống sẽ được cấu hình. Ví dụ, file UNIX “.profile”, sẽ được thực hiện khi người dùng đăng nhập vào hệ thống, nó thiết lập biến hệ thống và biến môi trường theo sở thích của người dùng, những hệ thống như vậy được gọi là có khả năng thích ứng, khi mà người dùng có thể thích ứng môi trường làm việc của anh ta phù hợp với sở thích. Điều này làm tăng tính linh hoạt của hệ thống nhưng lại đặt trách nhiệm tùy biến vào người dùng. Kết quả của điều này là người dùng chỉ truy cập được vào hệ thống mặc định lúc họ cần tính linh hoạt nhất: Đó là, khi họ lần đầu tiên bắt đầu. Và chỉ về sau họ mới biết cách xây dựng một mô hình cần thiết.

Tiếp cận thứ ba để cung cấp cho hệ thống một mô hình của người dùng là nhờ hệ thống xây dựng và duy trì một mô hình người dùng dựa trên những dữ liệu thu lượm trong lúc quan sát sự tương tác của người dùng. Mô hình này sau đó được tra cứu khi cần thiết. Cách tiếp cận tự động này đối với mô hình hoá người dùng cũng có vấn đề về thời gian cài đặt được yêu cầu, trong thời gian mà người dùng có hệ thống mặc định, nhưng trách nhiệm xây dựng mô hình lại không phải là của người dùng. Có rất nhiều ý kiến đưa ra để làm sao giải quyết vấn đề với thời gian cài đặt, trong đó có cả cho phép người dùng chọn một mô hình mặc định ban đầu, và xây dựng một mô hình dựa trên những hoạt động sử dụng trước đó, ví dụ như là chơi trò chơi. Với ý kiến đầu thì thật khó hiểu bởi vì nó khiến người dùng phải chọn mô hình có thể di chuyển sang lĩnh vực thực sự. Cách tiếp cận phổ biến nhất vẫn là cung cấp một mô hình mặc định cơ bản lúc ban đầu và cho phép người dùng có thể cập nhật nó một cách nhanh chóng. Mô hình mặc định có thể dựa trên những kết quả thí nghiệm và quan sát được thu lượm từ những đánh giá.

Vậy thì mô hình người dùng được xây dựng và duy trì như thế nào? Có rất nhiều cách tiếp cận để giải quyết vấn đề này. Một số đã định lượng kinh nghiệm người dùng hoặc là phân loại người dùng thành những mẫu; một số khác lại so sánh hành vi của người dùng

Chương 11: Thiết kế trợ giúp

với các chỉ tiêu; số khác lại dùng một danh mục các lỗi đã biết và so sánh hành động của người dùng với chúng.

Sự định lượng: Đây là một trong số những cách tiếp cận đơn giản nhất để mô hình hoá người dùng. Hệ thống nhận ra nhiều mức độ của sự tinh thông, và nó sẽ đáp lại theo những cách khác nhau. Người dùng sẽ được đặt vào một trong những mức đó, và nó sẽ chỉ thay đổi trong phạm vi những mức đó, dựa trên phương pháp đo định lượng sự tinh thông tại thời điểm đó. Những hành động khác nhau được gán với một trọng số, và người dùng sẽ được tính điểm theo trọng lượng của những hoạt động mà anh ta tham gia. Nếu điểm số vượt qua một ngưỡng nào đó, người dùng sẽ được di chuyển đến một mức tinh thông khác và khi đó hệ thống sẽ thích ứng theo đó.

Cách tiếp cận này đơn giản và đo được người dùng tại những mức rất nhỏ. Tuy nhiên, nó chỉ hiệu quả đối với sự thích ứng đơn giản. Ví dụ, phương pháp được sử dụng bởi Mason để thích ứng sự trình diễn của dấu nhắc lệnh đối với mức độ kinh nghiệm của người dùng. Hệ thống sử dụng một tập các luật được gọi khi mức độ tinh thông của người dùng thay đổi. Ví dụ,

Chuyển từ mức độ 1 tới mức độ 2

nếu

Hệ thống được sử dụng nhiều hơn 2 lần (0,25)

Câu lệnh x và y được sử dụng hiệu quả (0,20)

Trợ giúp không được truy cập trong phiên này (0,25)

Hệ thống được sử dụng 5 ngày trước.

Hệ thống như vậy chỉ có thể đưa ra được một sự ước lượng xấp xỉ về sự tinh thông của người dùng, nhưng tại cùng một lúc nó lại cần phân tích rất ít để đưa ra được những thông tin cần thiết từ bản ghi hệ thống.

Mẫu sẵn: Một cách tiếp cận khác trong việc mô hình hoá người dùng tự động là dùng mẫu sẵn. Không xây dựng một mô hình chính xác của người dùng, mà hệ thống phân loại người dùng như là một thành viên của danh mục người dùng đã biết gọi là những mẫu sẵn. *Những mẫu sẵn dựa trên những đặc tính của người dùng, nó có thể đơn giản như là một sự khác nhau giữa người dùng chuyên nghiệp và người dùng chưa kinh nghiệm, hoặc có thể phức tạp hơn, như là xây dựng một mẫu tinh vi dựa trên nhiều thông tin. Có rất nhiều cách để xây dựng nên những mẫu sẵn. Cách đầu tiên là sử dụng những thông tin như là sự sử dụng câu lệnh và những lỗi để phân loại ra thành những kiểu người dùng khác nhau, sau đó dùng những quy luật để xác định mẫu mà người dùng thuộc về. Cách tiếp cận khác là sử dụng cơ chế học, như là mạng neuron, để biết được những ví dụ của những kiểu hành vi người dùng khác nhau, và sau đó phân loại người dùng theo những mẫu gần nhất mà hệ thống đã học được. Những mẫu này rất hữu ích bởi vì nó biểu diễn người dùng tại mức rất nhỏ tại đó đa số hệ thống trợ giúp thích ứng đều làm việc, và không tạo ra một mô hình*

Chương 11: Thiết kế trợ giúp

phức tạp mà lại không được tận dụng hoàn toàn. Cuối cùng, nếu chỉ với những thông tin về cách người dùng tương tác với hệ thống, thì không đủ để suy ra nhiều về anh ta. Tuy nhiên, những gì được suy ra có thể là chính xác là những gì được yêu cầu để cung cấp một mức độ trợ giúp cần thiết.

Mô hình phụ: Đây là một mô hình được lý tưởng hoá, theo cách của một người dùng thành thạo, được xây dựng và hành vi của từng cá nhân người dùng được so sánh với nó. Kết quả đặc điểm người dùng có thể biểu diễn cả đặc điểm chung giữa hai mô hình và cả sự khác nhau. Một thuận lợi của phong cách này là nó cho phép một mức độ nào đó của hoạt động phân tích trên phần của hệ thống. Hệ thống không chỉ biết được những gì người dùng đang làm mà nó còn có được một sự trình bày về hành vi tối ưu nhất. Điều này cung cấp một tham chiếu để đo được sự thực hiện của người dùng, và nếu người dùng không thực hiện tiến trình tối ưu của hành động, thì nó sẽ đưa ra một chỉ định về kiểu của trợ giúp hoặc là lời gợi ý được yêu cầu.

Một cách tiếp cận tương tự được sử dụng trong mô hình dựa trên lỗi. Ở đó, hệ thống lưu trữ một bản ghi các lỗi của người dùng đã được biết, và những hành vi thực của người dùng sẽ được so sánh với các lỗi này. Nếu hành vi này hợp với một lỗi trong danh mục, thì một hành động cứu chữa sẽ được thay thế.

Các lỗi tiềm tàng có thể được so khớp khi được thực thi phần nào và trợ giúp được đưa ra cho phép người dùng tránh lỗi đó, hoặc là phục hồi lỗi một cách nhanh chóng. Kiểu mô hình hoá này thích hợp với những hệ thống giảng dạy thông minh ở đó những thông tin phân tích được yêu cầu để mà quyết định tiếp tục giảng dạy như thế nào.

11.4.2. Biểu diễn tri thức: mô hình hoá nhiệm vụ và lĩnh vực

Tất cả các hệ thống trợ giúp thích ứng đều phải có một số hiểu biết về hệ thống, để có thể cung cấp những lời khuyên thích hợp. Những hiểu biết này có thể là mục đích câu lệnh, những lỗi chung và những tác vụ chung. Tuy nhiên, một số hệ thống trợ giúp lại cố gắng xây dựng một mô hình tác vụ hoặc dự định hiện thời của người dùng. Động cơ đằng sau điều này là người dùng bị ràng buộc với một tác vụ giải quyết vấn đề nào đó, và yêu cầu trợ giúp tại thời điểm đó. Trợ giúp tổng quát, thậm chí là thích ứng với sự tinh thông và sở thích của người dùng, vẫn là không đủ.

Một cách tiếp cận chung cho vấn đề này là biểu diễn những tác vụ của người dùng bằng các thuật ngữ của dãy các lệnh được yêu cầu thực thi các tác vụ này. Khi người dùng làm việc, những câu lệnh được sử dụng được so sánh với các dãy tác vụ được lưu và dãy thích hợp được phục hồi. Nếu dãy lệnh của người dùng không hợp với một tác vụ được công nhận, thì trợ giúp được đưa ra.

Tuy là một ý tưởng hấp dẫn, nhưng sự nhận ra tác vụ là khó hiểu. Trong những lĩnh vực rộng lớn, thì không thể nào, phương pháp có thể để đạt được mục đích người dùng có thể lại được biểu diễn. Nhiều người dùng có thể tiếp cận một tác vụ theo cách không chuẩn, và việc luận ra được ý định của người dùng từ cách sử dụng câu lệnh không phải là vấn đề nhỏ.

11.4.3. Biểu diễn tri thức: mô hình hoá chiến lược tư vấn

Lĩnh vực thứ ba của sự trình diễn tri thức thành thạo thuộc về trợ giúp thích ứng là các chiến thuật cố vấn hoặc hướng dẫn mô hình hoá cung cấp một hệ thống trợ giúp với kiểu thông tin này cho phép nó không chỉ chọn ra những lời khuyên thích hợp cho người dùng mà còn sử dụng một phương pháp thích hợp để đưa lời khuyên.

Như chúng ta đã thấy, người dùng yêu cầu những kiểu trợ giúp khác nhau phụ thuộc vào tri thức và hoàn cảnh của họ. Đó có thể là những lời nhắc nhở, hay trợ giúp theo tác vụ và trợ giúp với mục đích giảng dạy. Những kiểu này có thể luận ra ý định của một người đang cần trợ giúp và đưa ra lời khuyên ở mức độ đó hoặc là cung cấp nhiều giải pháp cho vấn đề của anh ta. Hay theo cách khác, những trợ giúp kiểu này sẽ cố gắng đặt vấn đề vào trong một ngữ cảnh cụ thể và cung cấp những giải pháp mẫu cho ngữ cảnh đó.

Có rất ít hệ thống trợ giúp thích ứng cố gắng mô hình hoá chiến thuật tư vấn, chúng chỉ đưa được ra một số lựa chọn giới hạn. Một cách lý tưởng, nếu như một hệ thống trợ giúp có nhiều chiến thuật cung cấp trợ giúp. Tuy nhiên rất khó để biết được chiến thuật nào thì phù hợp với ngữ cảnh nào. Tuy nhiên, người thiết kế hệ thống trợ giúp thích ứng cũng nên nghĩ một chút một về chiến thuật tư vấn được sử dụng.

Với hệ thống trợ giúp EuroHelp, nó dùng mô hình “giáo viên – học sinh” ở đó hệ thống được khảo sát được coi như là một giáo viên theo dõi người dùng (học sinh) làm việc và đưa ra lời khuyên trong kiểu cách vai kẻ vai.

11.4.4 Các kỹ thuật biểu diễn tri thức

Tất cả các cách tiếp cận mà chúng ta đã trình bày dựa vào những kỹ thuật trình diễn tri thức của trí tuệ nhân tạo. Tuy nhiên trong tài liệu này chúng ta sẽ không đi sâu về lĩnh vực trí tuệ nhân tạo. Ta chia kỹ thuật trình diễn tri thức thành bốn nhóm chính: dựa trên quy luật, cơ cấu, mạng, ví dụ.

Kỹ thuật dựa trên các quy luật: Tri thức được đại diện như một tập hợp những quy tắc và những sự việc được thông dịch bằng cách sử dụng một số cơ chế suy diễn nào đó. Quy luật logic cung cấp một cơ chế hiệu quả để đại diện cho những thông tin tường thuật, trong khi những quy tắc sản xuất đại diện thông tin mang tính thủ tục. Kỹ thuật dựa trên các quy luật có thể được sử dụng trong lĩnh vực tương đối lớn và có thể đại diện những hoạt động để thực hiện cũng như kiến thức cho sự suy diễn. Một mô hình người dùng được thực hiện sử dụng những phương pháp trên dựa trên các quy luật có thể bao gồm những quy tắc của những mẫu sau:

```
If command is EDIT file1  
And  
last command is COMPILE file1  
Then  
task is DEBUG  
action is describe automatic debugger
```

Chương 11: Thiết kế trợ giúp

Kỹ thuật dựa trên cấu trúc: Những hệ thống trên nền cấu trúc được sử dụng để đại diện những tình huống thông thường xuất hiện và kiến thức mặc định. Một cấu trúc chứa đựng gắn nhãn những vị trí, những đặc tính liên quan đại diện. Mỗi vị trí có thể được gán một giá trị hoặc được cho một giá trị mặc định. Dữ liệu vào của người dùng được so sánh đối với những giá trị cấu trúc và một kết hợp thành công có thể gây ra hoạt động nào đó sẽ được giữ. Chúng hữu ích trong những lĩnh vực nhỏ. Trong mô hình hóa người dùng cấu trúc có thể đại diện trạng thái người dùng:

Độ thành thạo: chưa có kinh nghiệm

Lệnh : EDIT file1

Lệnh trước: COMPILE file1

Những lỗi trong phiên làm việc này : 6

Hoạt động : Mô tả trình gỡ rối tự động

Kỹ thuật dựa trên mạng lưới: Những mạng biểu diễn kiến thức về người dùng và hệ thống dưới dạng những mối quan hệ giữa những sự việc. Ví dụ điển hình nhất là mạng ngữ nghĩa học. Mạng lưới là một sự phân cấp và các nút con có thể kế thừa những thuộc tính liên quan đến nút cha. Điều này làm cho nó trở thành một sơ đồ trình bày một cách tương đối hiệu quả và hữu ích cho việc kết nối thông tin một cách rõ ràng. Những mạng có thể cũng sử dụng để liên kết những sự trình bày dựa trên cấu trúc. Ví dụ biên tập có thể được mở rộng bên trong một mạng ngữ nghĩa:

CC là một thể hiện (của) sự COMPILE

COMPILE là một lệnh

COMPILE có liên quan đến DEBUG

COMPILE có liên quan đến EDIT

Bộ gỡ lỗi tự động làm đơn giản việc Gỡ lỗi

Kỹ thuật dựa trên Ví dụ: Kỹ thuật dựa trên Ví dụ biểu diễn kiến thức hoàn toàn bên trong một cấu trúc quyết định của một hệ thống phân loại. Nó có thể là một cây quyết định, trong trường hợp một cách tiếp cận học cảm ứng như là ID3, hoặc những mối liên kết trong một mạng lưới trong trường hợp mạng neuron. Cấu trúc quyết định được xây dựng tự động dựa vào những ví dụ được giới thiệu tới người phân loại. Những người phân loại phát hiện ra những đặc tính lặp lại bên trong những ví dụ và có thể sử dụng những điều đó để phân loại những dữ liệu vào khác. Một ví dụ có thể là một dấu vết hoạt động của người dùng :

EDIT file1

COMPILE file1

Những câu lệnh này được sửa như một ví dụ (của) một tác vụ đặc biệt, ví dụ là
DEBUG

Một số điểm cần chú ý khi thiết kế hệ thống thích ứng

Sự trình bày kiến thức là vấn đề trung tâm trong những hệ thống giúp đỡ thích ứng, nhưng nó không phải không có những vấn đề. Kiến thức thường thì khó gọi ra, đặc biệt nếu một chuyên gia về lĩnh vực đó không có ở đây. Điều này còn đúng với tri thức về hành vi người dùng, do tính biến thiên của nó. Rất khó để bảo đảm tính chất toàn vẹn và tính chính xác cơ sở tri thức trong những hoàn cảnh này. Thậm chí nếu kiến thức sẵn có, số lượng tri thức được yêu cầu là quan trọng, làm cho trợ giúp thích ứng thành một sự lựa chọn đắt giá.

Vấn đề thứ hai là thông dịch thông tin một cách phù hợp. Mặc dù cơ sở tri thức có thể được cung cấp với tri thức chi tiết về những ngữ cảnh được mong đợi và lĩnh vực trước, trong lúc tương tác thông tin duy nhất mà sẵn có là các bản ghi hệ thống về các hoạt động của người dùng. Thông dịch những bản ghi hệ thống dịch rất khó bởi vì nó đã bị loại đi nhiều ngữ cảnh và không có sự truy nhập tới dự định hoặc mục đích người dùng. Tuy nhiên, dữ liệu này không phải là tùy ý và chứa đựng những mẫu lặp lại của các hoạt động mà có thể được sử dụng với mỗi quan tâm để suy ra những thứ tự nhiệm vụ và tương tự. Tuy nhiên, nó chỉ cần phải được nhận thức rằng những điều đó chỉ biểu diễn những sự xấp xỉ.

11.5. Thiết kế hệ thống trợ giúp người dùng

Có nhiều cách cung cấp trợ giúp cho người dùng và nó phụ thuộc vào người thiết kế quyết định xem cái gì là thích hợp nhất cho bất kỳ hệ thống nào cho trước. Tuy vậy, có rất nhiều thứ mà người thiết kế nào cũng cần phải tính đến. Trước hết, thiết kế của trợ giúp người dùng không nên được xem như là một “phụ kiện để mở rộng” tới thiết kế hệ thống. Hay nói cách khác hệ thống trợ giúp cần phải được thiết kế toàn bộ với phần còn lại của hệ thống. Như vậy các kỹ thuật mô hình hóa và kỹ thuật phân tích yêu cầu được sử dụng cho việc thiết kế hệ thống phải được áp dụng cho việc thiết kế hệ thống trợ giúp. Hai là, người thiết kế cần phải xem xét nội dung của sự trợ giúp và ngữ cảnh trong đó nó sẽ được sử dụng trước khi quyết định công nghệ cần thiết. Rõ ràng, công nghệ sẵn có là một vấn đề quan trọng. Tuy nhiên, việc tập trung vào nhiệm vụ và người dùng sẽ giúp cho việc xác định rõ ràng kiểu trợ giúp được yêu cầu đối với sự ràng buộc của những tài nguyên về kỹ thuật. Nhìn quá trình theo cách ngược lại có thể ngăn người thiết kế thấy được công nghệ bên ngoài mà anh ta đã quen với.

11.5.1. Mô thức trình diễn

Trợ giúp được yêu cầu như thế nào?

Quyết định đầu tiên mà người thiết kế phải làm là sự giúp đỡ sẽ được truy nhập bởi người dùng như thế nào. Có một số sự lựa chọn cho vấn đề này. Trợ giúp có thể là một lệnh, một nút, một hàm có thể nào chuyển trạng thái “bật” “tắt”, hoặc là một ứng dụng riêng biệt. Một lệnh (thông thường) yêu cầu người dùng chỉ rõ một chủ đề, và bởi vậy đưa ra một số tri thức, nhưng có thể không nhất quán với các giao diện còn lại. Một nút bấm trợ giúp có thể sẵn sàng được truy cập mà không can thiệp vào ứng dụng đang chạy, nhưng lại không cung cấp thông tin đặc biệt cho những nhu cầu của người dùng. Tuy nhiên, nếu nút trợ giúp là

Chương 11: Thiết kế trợ giúp

một phím hoặc nút chuột, thì nó có thể hỗ trợ tính nhạy cảm về ngữ cảnh. Hàm giúp đỡ thì linh hoạt một khi nó có thể được kích hoạt khi nào cần thiết và bị vô hiệu hóa khi nào không. Ứng dụng riêng biệt cho phép tính linh hoạt và những nhiều kiểu trợ giúp đỡ nhưng lại can thiệp vào ứng dụng hiện thời của người dùng.

Sự trợ giúp được trình bày như thế nào?

Quyết định chính thứ hai mà người thiết kế phải làm là trợ giúp sẽ được xem bởi người sử dụng. Trong một hệ thống của số nó có thể có là cửa sổ mới. Trong những hệ thống khác nó có thể sử dụng toàn bộ màn ảnh hoặc bộ phận của màn hình. Nói cách khác, những gợi ý có thể được những hộp pop-up hoặc ở tại mức hàng lệnh. Kiểu biểu diễn mà thích hợp phụ thuộc phần lớn vào mức của trợ giúp được yêu cầu và không gian mà nó yêu cầu. Vài hệ thống trợ giúp chủ động cung cấp những gợi ý trực quan khi chúng được yêu cầu (ví dụ, một biểu tượng có thể được làm cho sáng lên) - điều này cho phép người dùng tùy ý lựa chọn có cần gợi ý không mà không bắt buộc anh ta phải từ bỏ hoặc ngắt công việc của anh ta.

Sự trình diễn hiệu quả của trợ giúp:

Màn hình trợ giúp và tài liệu cần phải được thiết kế tương tự cách một giao diện được thiết kế, phải tính đến những khả năng và yêu cầu nhiệm vụ của người dùng. Cho dù công nghệ nào được sử dụng để cung cấp trợ giúp thì cũng có vài nguyên lý để viết và biểu diễn nó một cách có hiệu quả. Trợ giúp và các tài liệu giảng dạy phải được viết trong ngôn ngữ sáng sủa, quen thuộc, tránh tiếng lóng càng nhiều càng tốt. Nếu những tài liệu giấy tồn tại, thì thuật ngữ học cần phải là nhất quán với tài liệu trực tuyến. Những tài liệu hướng dẫn yêu cầu ngôn ngữ hướng dẫn và một hệ thống trợ giúp cần phải nói người dùng làm sao để sử dụng hệ thống hơn là chỉ mô tả hệ thống. Và phải xác định xem cái gì người dùng muốn biết trước tiên. Ví dụ, một thông báo giúp đỡ sự sử dụng cửa sổ.

Để đóng cửa sổ, hãy đặt con trỏ chuột về cái hộp trên đỉnh ở góc tay trái của cửa sổ và kích nút chuột.

Hơn là

Cửa sổ có thể được đóng bằng cách đưa chuột về cái hộp trên đỉnh ở góc tay trái của cửa sổ và kích nút chuột.

Một ngoại lệ cho điều này là trong tài liệu với ý định không chỉ hướng dẫn người dùng làm sao để sử dụng hệ thống mà còn ghi lại một sự mô tả đầy đủ chức năng hệ thống. Tuy nhiên, tài liệu cần phải có sao cho thông tin có thể sẵn sàng được truy cập, và cần phải biểu diễn cả thông tin mang tính hướng dẫn lẫn miêu tả một cách rõ ràng. Cách trình bày về mặt vật lý của tài liệu có thể có một số khác nhau về tính năng của nó. Những văn bản lớn có thể gây khó khăn cho việc đọc trên màn hình. Như vậy có thể chia tài liệu thành phần rõ ràng. Những mục, hoặc việc sử dụng công nghệ siêu văn bản để tổ chức văn bản bằng cách sử dụng các siêu liên kết để liên kết chúng. Một cách tốt nhất là ta cung cấp những thông tin quan trọng, còn những thông tin thêm thì có thể sẵn sàng khi được yêu cầu. Có thể sắp đặt

Chương 11: Thiết kế trợ giúp

một hệ thống trợ giúp có thứ bậc nơi mỗi lớp trong sự phân cấp cung cấp chi tiết ngày càng sâu hơn, hoặc đơn giản bởi việc sử dụng cách trình bày cẩn thận. Chỉ số có thể được sử dụng để tóm lược những chủ đề sẵn có nhưng cần phải được tổ chức để phản ánh những mối quan hệ chức năng giữa những đề tài hơn là theo văn chữ cái. Sự nhất quán cũng là vấn đề quan trọng ở đây - mỗi chủ đề trong tài liệu cần phải được mô tả phải sử dụng cùng khuôn dạng để người dùng biết mà để tìm một loại thông tin nào đó. Tài liệu và trợ giúp có thể chứa đựng những định nghĩa, những sự mô tả, những ví dụ, những chi tiết của những thông báo lỗi, những tùy chọn và chi dẫn.

11.5.2. Mô thức cài đặt

Theo những vấn đề trình diễn, người thiết kế phải đưa ra những quyết định cài đặt. Chúng có thể bị ràng buộc bởi những sự ràng buộc vật lý, những cái khác bởi những sự lựa chọn được làm quan tâm tới những yêu cầu của người dùng cho trợ giúp. Chúng ta đã xem xét trợ giúp có thể được yêu cầu như thế nào và nó được biểu diễn như thế nào với người dùng. Rõ ràng mỗi quyết định này kéo theo những câu hỏi về sự thi hành : trợ giúp sẽ là một lệnh hệ điều hành, một siêu lệnh hoặc một ứng dụng? Những sự ràng buộc vật lý nào làm cho bộ máy áp đặt dưới dạng không gian màn hình, khả năng bộ nhớ và tốc độ? Tốc độ là một sự xem xét rất quan trọng, một khi một thời gian phản ứng chậm không thể chấp nhận được làm cho hệ thống không sử dụng được cho dù nó được thiết kế rất tốt. Tốt hơn là cung cấp phương tiện trợ giúp đơn giản mà trả lời nhanh hơn một hệ thống tinh vi mất những nhiều thời gian mới cung cấp được một giải pháp.

Vấn đề khác mà người thiết kế phải quyết định là dữ liệu trợ giúp sẽ được xây dựng như thế nào: Trong một tệp, một sự phân cấp tệp, một cơ sở dữ liệu? Điều này sẽ phụ thuộc vào kiểu trợ giúp được yêu cầu, nhưng với bất kỳ cấu trúc nào thì cũng cần linh hoạt và có tính mở rộng - những hệ thống thì không bất biến và khi đó có những chủ đề sẽ tất yếu cần thêm vào hệ thống trợ giúp. Cấu trúc dữ liệu sử dụng sẽ xác định chiến lược, kiểu tìm kiếm hoặc sự định vị. Người dùng có thể cũng muốn làm một bản cứng sao chép về phần nào đó của hệ thống để sau này đọc.

Cuối cùng người thiết kế cần phải xem xét những tác giả của tài liệu trợ giúp cũng như những người dùng của nó. Có vẻ là, dù người thiết kế viết những văn bản giúp đỡ ban đầu, thì chúng sẽ được mở rộng trước những tác giả khác ở những thời gian khác nhau. Những quy ước rõ ràng và những ràng buộc về sự trình bày và sự thực thi đã giúp cho việc thêm các tài liệu mới được đơn giản.

11.6. Ví dụ thiết kế một hệ trợ giúp dạng HTML Help

Để tạo một chương trình trợ giúp dạng này, người ta tiến hành theo ba bước sau:

- 1) Tạo các Help topic (chủ đề Help).
- 2) Xác định các cửa sổ để hiển thị các Help topic.
- 3) Thiết kế phương pháp định hướng các Help topic.

1. Tạo các Help topic

Help topic (chủ đề Help) là những màn hình thông tin tách biệt. Chúng thường rất cô đọng và là đơn vị tổ chức cơ bản trong một hệ trợ giúp, chúng cũng chứa đựng tất cả những thông tin mà người dùng tìm kiếm.

Việc tạo các Help cũng giống như tạo nội dung hiển thị trong cửa sổ Help. Dưới đây là một số loại Help topic.

Welcome Topic: đây là chủ đề đầu tiên trong một hệ trợ giúp, nó biểu diễn mục đích tổng thể của một hệ trợ giúp. Ví dụ về Welcome topic là “Welcome”, hoặc “Getting Started”.

Overview style topic: cung cấp thông tin mang tính khái niệm và nền tảng về một chủ đề. Loại Topic này thường bắt đầu bằng một đầu đề có dạng là một cụm danh từ.

Procedure style topic: biểu diễn một chuỗi các bước giúp người dùng hoàn thành một nhiệm vụ cụ thể. Nhiều các topic loại này bắt đầu với một tiêu đề mô tả một vài loại hoạt động, ví dụ: “Creating a Topic”.

Thường thì các procedure topic bắt đầu bằng một giới thiệu khái niệm ngắn gọn, theo sau là một chuỗi các bước. Khái niệm tổng thể có thể được theo sau hoặc thay thế bởi một dạng nguyên thể như “To create a Topic inside the Project Tab”. Đôi khi, các bước này được chia nhỏ thành nhiều hơn một chuỗi. Một chuỗi mới có thể bắt đầu bằng một đầu đề con như “To create a New Topic while inside the TOC composer”.

Definition style topics: là các mô tả ngắn gọn thường được hiển thị trong một cửa sổ pop-up. Chúng giúp người dùng hiểu về các thuật ngữ trong một Help topic lớn hơn mà không làm người dùng phải rời topic họ đang xem và làm mất tập trung của họ.

What’s This? Style topics: là những topic pop-up nhỏ người dùng có thể hiển thị bằng cách kích chuột vào một tính năng giao diện trong một ứng dụng để cung cấp một mô tả ngắn gọn về tính năng. Một What’s This? Style topic không được thiết kế để hướng nhiệm vụ, mà nó đơn giản chỉ là mô tả các chức năng cụ thể trên giao diện ứng dụng.

Một số các loại chủ đề khác: bao gồm các thông điệp lỗi (error message), các chủ đề xử lý sự cố (troubleshooting), các chủ đề hiển thị trong các ứng dụng trình diễn đa phương tiện.

2. Xác định các cửa sổ để hiển thị các help topic

Xác định các cửa sổ và màn hình Help là bước thứ hai để phát triển một hệ thống Help. Sau khi tạo xong các chủ đề, bạn cần quyết định loại cửa sổ nào sẽ chứa những chủ đề này. Khi thiết kế cửa sổ và màn hình, cần phải xem xét các ý kiến sau:

- Không bắt buộc các cửa sổ Help cảm ngữ cảnh (context-sensitive) phải hoàn thành quá nhiều nhiệm vụ cho màn hình thực được dùng chung trong toàn bộ ứng dụng. Người dùng thường cảm thấy khó chịu khi họ phải thay đổi kích

Chương 11: Thiết kế trợ giúp

thước các cửa sổ Help vì chúng che phủ lên màn hình ứng dụng.

- Khi sử dụng các cửa sổ Help cảm ứng cảnh, cố gắng làm cho các cửa sổ này luôn ở trên các cửa sổ khác để người dùng không phải lật giờ cửa sổ Help của họ lên trên sau khi thực hiện xong một bước trong thủ tục.
- Không nhét quá nhiều thông tin vào một màn hình Help. Cho phép đủ các khoảng trống trong các chủ đề và tạo ra khoảng cách giữa cửa sổ và viền bảng và văn bản. Việc này làm cho người dùng cảm thấy dễ đọc hơn.
- Tránh dùng hình ảnh nền, hình mờ, màu sắc làm cho văn bản chủ đề Help khó đọc. Thay vào đó, chọn các font và màu sắc hài hoà như văn bản màu đen hoặc xanh nước biển trên nền màn hình màu trắng, màu kem hoặc những màu nhạt.
- Khi sử dụng hình nền hoặc hình chìm, đảm bảo rằng màu nền mặc định của cửa sổ phải hoà hợp với ảnh nền hoặc hình chìm đó. Khi thể hiện một chủ đề với một hình ảnh nền hoặc hình chìm, thường phải mất một khoảng thời gian để tải hình ảnh. Nếu màu ngầm định của màn hình Help khác hoàn toàn với ảnh nền, một tia sáng sẽ xuất hiện gây mất tập trung trước khi ảnh nền xuất hiện.
- Cố gắng đơn giản thiết kế Help. Trong phần lớn trường hợp, bạn cần rất ít các mẫu trong một hệ thống Help. Giữ cho quy ước về kiểu mẫu nhất quán. Tạo các khoảng trống theo chiều dọc và căn lề để tổ chức các thông tin trợ giúp sao cho dễ đọc.
- Dùng các bảng khi tổ chức các thông tin phức tạp và khi so sánh.
- Dùng các định nghĩa pop-up, cũng như các văn bản mở rộng hoặc thả xuống để giải thích các thuật ngữ sử dụng trong các cửa sổ chính và cửa sổ thứ cấp. Xem xét việc gom các định nghĩa ngắn gọn trong một glossary hoặc trong một tab glossary.

Các loại cửa sổ

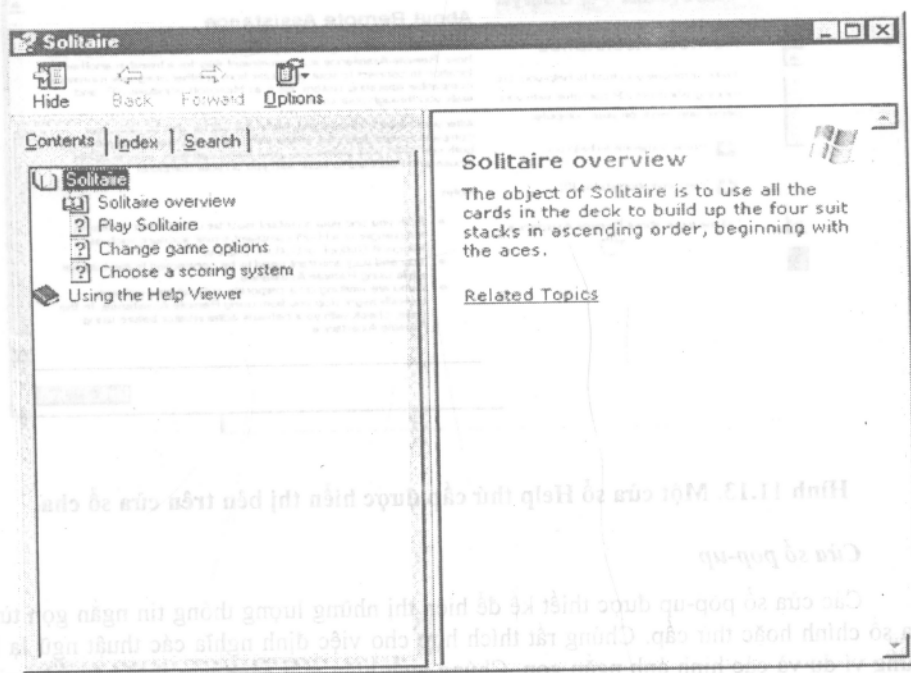
Có ba loại cửa sổ trong HTML Help: các cửa sổ chính, cửa sổ thứ cấp, và cửa sổ popup. Các cửa sổ chính và cửa sổ thứ cấp được biết tới như các cửa sổ tĩnh. Chúng vẫn tồn tại trên màn hình của người dùng cho đến khi họ đóng lại. Các cửa sổ thứ cấp mà xuất hiện từ các liên kết trong cửa sổ chính sẽ tự đóng khi cửa sổ chính được đóng lại. Các cửa sổ pop-up được biết tới như các cửa sổ tạm thời, chúng vẫn tồn tại trên màn hình của người dùng khi chúng vẫn còn focus. Chúng được đóng một cách tự động.

Cửa sổ chính

Các cửa sổ chính điển hình bao gồm một lượng lớn các văn bản. Dù người tạo Help có thể cấu hình cửa sổ chính theo bất kỳ cách nào, chúng thường bao gồm một thanh thực

Chương 11: Thiết kế trợ giúp

đơn (menu bar) và một thanh công cụ (toolbar). Người tạo Help có thể thiết kế một kích thước mặc định và vị trí cho tất cả các cửa sổ Help chính, mặc dù vậy, người dùng có thể thay đổi lại kích thước hoặc di chuyển các cửa sổ Help chính và thứ cấp theo cách mà họ muốn. Bạn có thể tổ chức các cửa sổ HTML Help để chúng mặc định về kích cỡ và vị trí xuất hiện do người dùng định nghĩa sau khi cửa sổ được đóng lại.



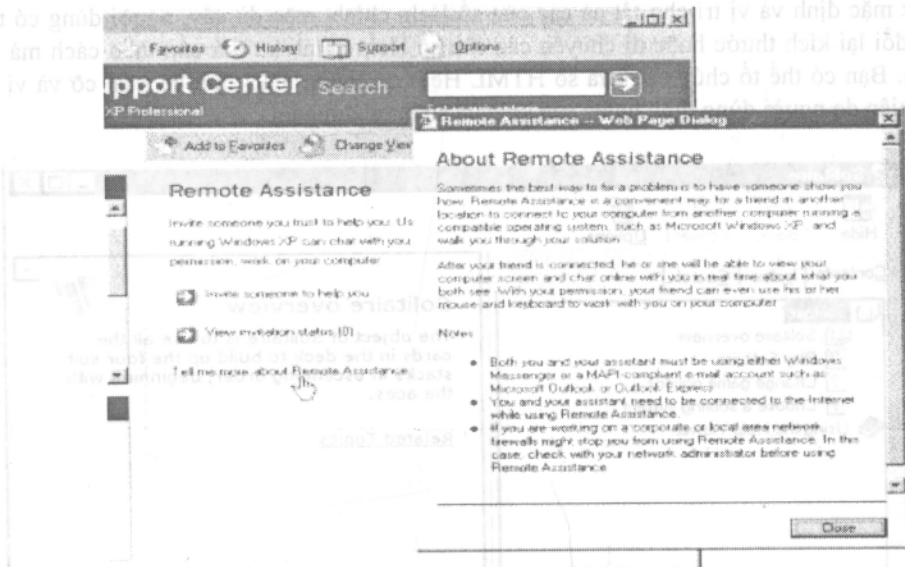
Hình 11.12. Một cửa sổ Help chính.

Việc thiết kế một kích thước và vị trí mặc định của các cửa sổ Help có thể khó giải quyết vì 2 lý do. Một mặt, bạn sẽ không muốn che giấu quá nhiều sản phẩm phần mềm. Mặt khác, các cửa sổ nhỏ hơn đòi hỏi việc cuộn cửa sổ và người dùng nói chung thì thích những cửa sổ mà không cần phải cuộn nhiều. Đây là mâu thuẫn mà bạn phải cố gắng điều hoà để tạo ra một thiết kế cửa sổ tối ưu.

Khi sử dụng các bảng, một điều rất quan trọng là phải tránh cho người dùng không phải cuộn ngang. Ngoài ra, bạn muốn các cửa sổ phải đủ lớn để chứa được tất cả các hình ảnh đồ họa của mình, thay vì bắt buộc người dùng cuộn các hình ảnh.

Cửa sổ thứ cấp

Cửa sổ thứ cấp xuất hiện cùng hoặc thay thế cho một cửa sổ Help chính. Bạn có thể sử dụng chúng cho nhiều mục đích. Tốt nhất là nên thiết kế các cửa sổ thứ cấp sao cho chúng không bao trùm lên phần ứng dụng mà chúng giải thích.



Hình 11.13. Một cửa sổ Help thứ cấp được hiển thị bên trên cửa sổ cha.

Cửa sổ pop-up

Các cửa sổ pop-up được thiết kế để hiển thị những lượng thông tin ngắn gọn từ một cửa sổ chính hoặc thứ cấp. Chúng rất thích hợp cho việc định nghĩa các thuật ngữ lạ hoặc những ví dụ và các hình ảnh ngắn gọn. Chúng xuất hiện một cách tạm thời bên trên cửa sổ lớn hơn. Bằng cách này, khi người dùng lựa chọn một liên kết pop-up chứa một thuật ngữ là, một văn bản hiển thị trong pop-up cho phép người dùng duy trì điểm nhìn của mình trong cửa sổ chính mà không phải ra khỏi chủ đề trong cửa sổ cha.

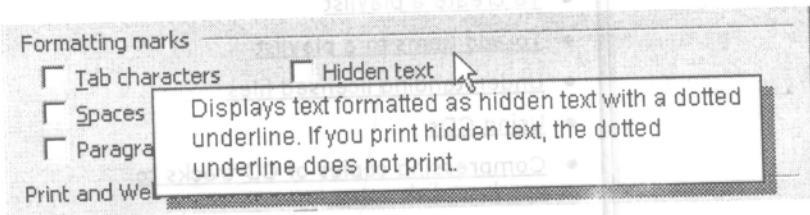
- Word or phrase prediction software, which allows users to type more quickly and with fewer keystrokes.
- Alternate input devices, such as single switch or puff and sip device

puff and sip device
 An assistive computer technology for people with mobility impairments. A puff and sip device is a head-mounted alternative to using the mouse. The device allows a user to move the mouse pointer without using his or her hands by puffing air into a tube.

Hình 11.14. Cửa sổ pop-up.

Chương 11: Thiết kế trợ giúp

Một loại khác của cửa sổ pop-up hay được sử dụng là loại cửa sổ “What’s This?”. Nó bao gồm một khối văn bản nhỏ trong một cửa sổ tạm thời.



Hình 11.5. Một mẫu cửa sổ Help pop-up.

3. Thiết kế phương pháp định hướng các Help topic

Đây là bước cơ bản cuối cùng để tạo ra một hệ trợ giúp. Các hệ trợ giúp dựa HTML đưa ra nhiều cách tiếp cận cho việc định hướng.

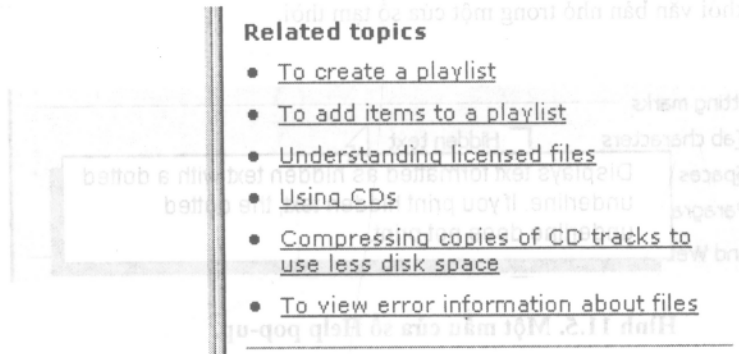
Các siêu liên kết dẫn tới các chủ đề liên quan (Hyperlinks to Related Topics): tương tự như những lời chỉ dẫn tham chiếu trong các tài liệu in nhưng nó nhanh hơn và thuận tiện hơn, và thường được ưa chuộng sử dụng. Hãy xem xét các ý kiến sau khi sử dụng các siêu liên kết:

- Tránh dùng các siêu liên kết nếu bạn có ý định làm việc với các điều khiển chủ đề liên quan (được đề cập đến ở phần dưới). Các siêu liên kết thường khó bảo trì hơn so với các điều khiển chủ đề liên quan.
- Không tạo ra các siêu liên kết vượt ra ngoài văn bản trong cùng một chủ đề khi chỉ cần một tham chiếu đơn là đủ. Tránh tạo ra nhiều hơn 4 hoặc 5 siêu liên kết tới các chủ đề liên quan. Nếu không, người dùng có thể sẽ bị quá tải với thông tin.
- Xem xét việc sử dụng các siêu liên kết để hiển thị thông tin bổ sung trong các cửa sổ pop-up hoặc các cửa sổ phụ thuộc. Bằng cách đó, các siêu liên kết của bạn sẽ không bỏ qua chủ đề hiện tại và đột ngột thay đổi khung nhìn của người dùng.
- Các siêu liên kết không được sâu quá ba mức.

Siêu liên kết và các điều khiển chủ đề liên quan:

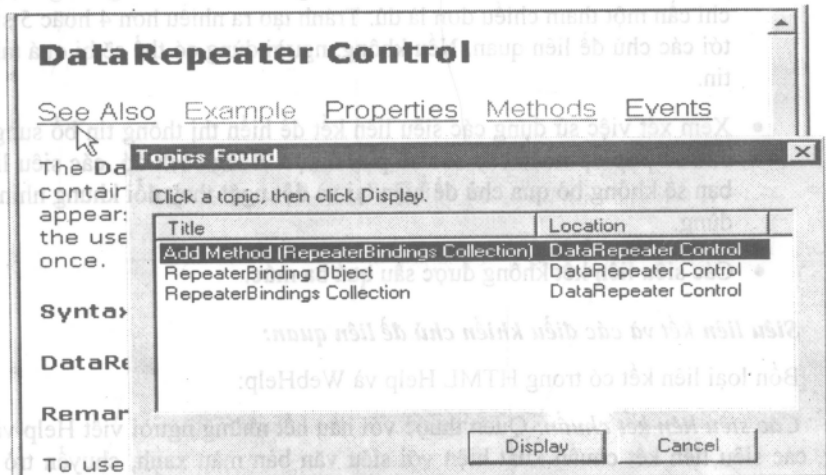
Bốn loại liên kết có trong HTML Help và WebHelp:

Các siêu liên kết chuẩn: Quen thuộc với hầu hết những người viết Help và người sử dụng, các siêu liên kết chuẩn xuất hiện với siêu văn bản màu xanh, chuyển trỏ chuột của người sử dụng thành hình bàn tay khi di chuyển qua siêu liên kết. Việc chọn một siêu liên kết sẽ truy nhập trực tiếp tới các chủ đề khác.



Hình 11.5. Danh sách các siêu liên kết tới các chủ đề liên quan.

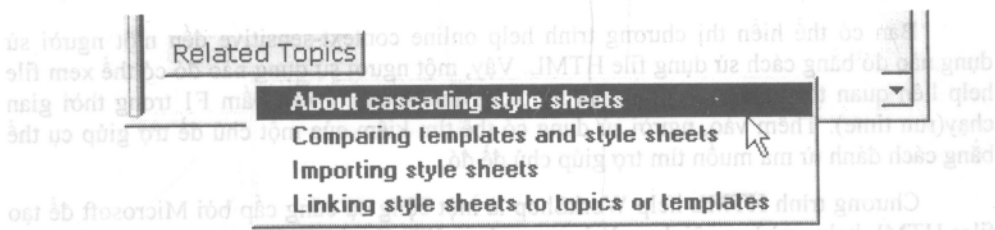
Các điều khiển *See Also*: Thay vì việc tạo ra rất nhiều các siêu liên kết để truy nhập tới các chủ đề liên quan, người viết Help có thể sử dụng một nút “See Also” hoặc một liên kết để chỉ ra một danh sách các chủ đề liên quan trong một hộp thoại hoặc một cửa sổ pop-up. Danh sách các chủ đề được tạo ra bằng cách liên kết các chủ đề với một từ khoá “See Also”, và sau đó tạo ra một nút hoặc liên kết hiển thị các chủ đề được liên kết với từ khoá. Điều khiển See Also rất linh hoạt: nếu xoá bỏ một chủ đề thuộc về một danh sách điều khiển See Also trong một hệ trợ giúp, tham chiếu tới chủ đề trong danh sách sẽ tự động được xoá bỏ. Điều này cho phép ta có thể copy và paste các nút điều khiển See Also vào nhiều chủ đề.



Hình 11. 16. Điều khiển See Also hiển thị một danh sách các chủ đề liên quan.

Các điều khiển liên kết từ khoá (Keyword Link Control): Giống như điều khiển See Also, điều khiển này giúp những người viết Help tránh phải tạo ra nhiều siêu liên kết tới các chủ đề liên quan khác trong cùng một chủ đề. Thay vào đó, điều khiển từ khoá Keyword Link giúp dễ dàng tạo ra một nút trong một chủ đề để kích hoạt hoặc là một cửa sổ pop-up hoặc là một hộp thoại để hiển thị nhiều siêu liên kết tới các chủ đề liên quan đó. Danh sách các chủ đề được dựa trên các chủ đề đã được liên kết với một điểm đầu vào (entry) hoặc từ khoá chỉ số nào đó. Điều khiển liên kết từ khoá cũng rất linh hoạt: nếu bạn xóa một chủ đề liên kết với một điều khiển liên kết từ khoá trong hệ thống trợ giúp, tham chiếu tới chủ đề cũng được tự động xóa bỏ khỏi danh sách của điều khiển liên kết từ khoá.

Các điều khiển chủ đề liên quan: Không giống như điều khiển liên kết từ khoá hay điều khiển See Also, điều khiển chủ đề liên quan cho phép người viết Help chọn các chủ đề liên quan một cách đơn lẻ để xuất hiện trong một cửa sổ pop-up hoặc một hộp thoại. Điều khiển này tạo ra các nút hoặc các liên kết trong các chủ đề để kích hoạt một danh sách các siêu liên kết tới các chủ đề liên quan.



Hình 11.17. Điều khiển các chủ đề liên quan hiển thị một danh sách các chủ đề liên quan.

- Bảng nội dung tóm tắt mà người sử dụng có thể chọn một chủ đề và xem nội dung ấy.
 - Tóm tắt Index mà người sử dụng có thể đánh từ khoá và chủ đề liên quan sẽ hiển thị.
- Hệ thống trợ giúp này sẽ chứa trang chủ đề để:
- Thêm vào chỉ tiết của một khách hàng mới.
 - Sửa đổi chỉ tiết của một khách hàng tồn tại trong hệ thống.
 - Xóa chỉ tiết về một khách hàng tồn tại.
 - Biểu hướng qua các chỉ tiết của khách hàng tồn tại.
3. Tạo hệ thống trợ giúp
- Thực hiện các bước sau:
1. Mở Chương trình HTML Help Workshop. Giao diện của HTML help Workshop sẽ hiển thị như hình sau:

Phụ lục A

TẠO HỆ TRỢ GIÚP DẠNG HTML HELP

1. Xác định cơ chế để đưa ra trợ giúp

Chương trình help là một công cụ tiện ích của các ứng dụng. Một ứng dụng coi như là có hiệu quả nếu có chứa một hệ thống trợ giúp giải thích. Khi sử dụng một ứng dụng, một người sử dụng có thể cần chương trình help để thực hiện một thao tác hoặc yêu cầu thông tin về một nút hoặc một hộp thoại.

HTML help biểu hiện nội dung chương trình help từ một file HTML. Khi người sử dụng nhấn phím F1 trong thời gian chạy (run time) thì trang HTML chứa nội dung help được hiển thị.

Bạn có thể hiển thị chương trình help online context-sensitive đến một người sử dụng nào đó bằng cách sử dụng file HTML. Vậy, một người sử dụng nào đó có thể xem file help liên quan tới một hành động mà họ đang thực hiện chỉ cần bấm F1 trong thời gian chạy(run time). Thêm vào, người sử dụng có thể tìm kiếm của một chủ đề trợ giúp cụ thể bằng cách đánh từ mà muốn tìm trợ giúp chủ đề đó.

Chương trình HTML help Workshop là một công cụ cung cấp bởi Microsoft để tạo files HTML help, mà lưu nội dung help trong dạng HTML.

2. Dự thảo thiết kế hệ thống trợ giúp

Chương trình help của module Customer Details sẽ chứa trang chủ đề. Hệ thống trợ giúp sẽ cung cấp người sử dụng hai tầm nhìn :

- Bảng nội dung tầm nhìn mà người sử dụng có thể chọn một chủ đề và xem nội dung ấy.
- Tầm nhìn Index mà người sử dụng có thể đánh từ khoá và chủ đề liên quan sẽ hiển thị.

Hệ thống trợ giúp này sẽ chứa trang chủ đề đề:

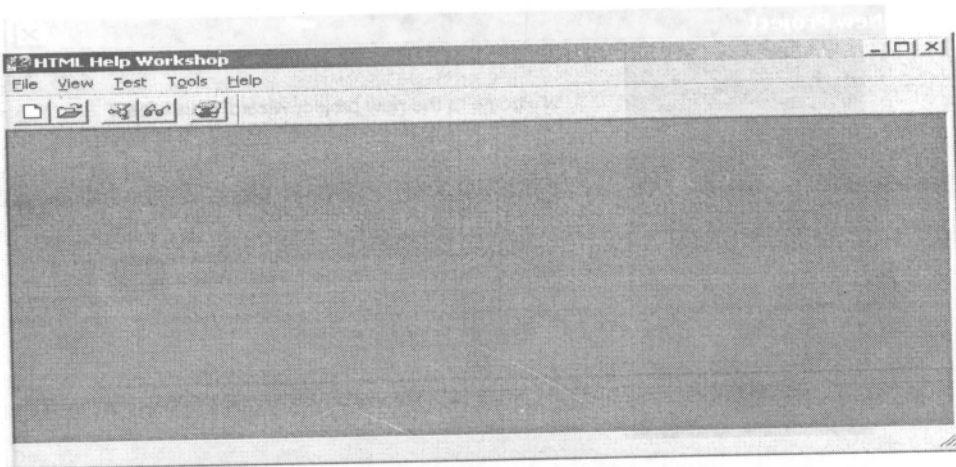
- Thêm vào chi tiết của một khách hàng mới.
- Sửa đổi chi tiết của một khách hàng tồn tại trong hệ thống.
- Xoá chi tiết về một khách hàng tồn tại
- Điều hướng qua các chi tiết của khách hàng tồn tại.

3. Tạo hệ thống trợ giúp

Thực hiện các bước sau:

1. Mở Chương trình HTML Help Workshop. Giao diện của HTML help Workshop sẽ hiển thị như hình sau:

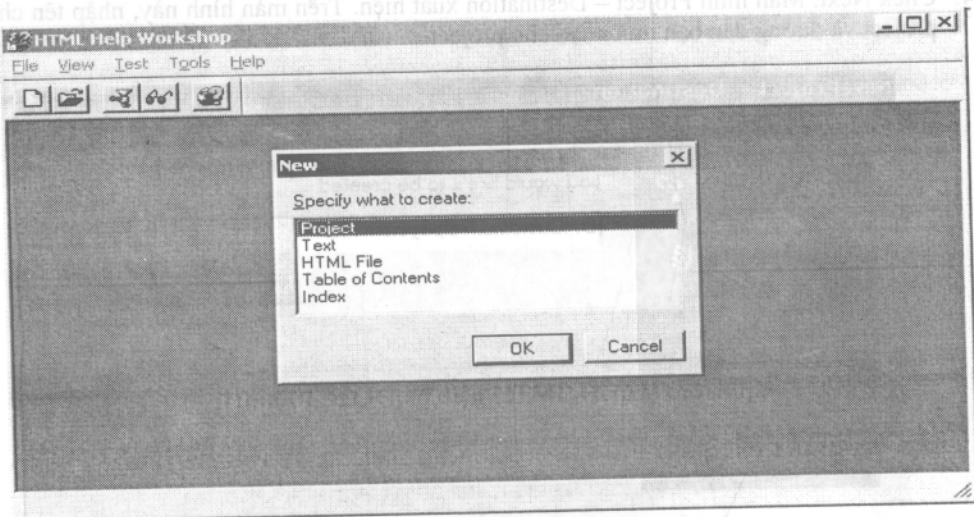
Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop



Hình PLA.1. Giao diện HTML Help Workshop.

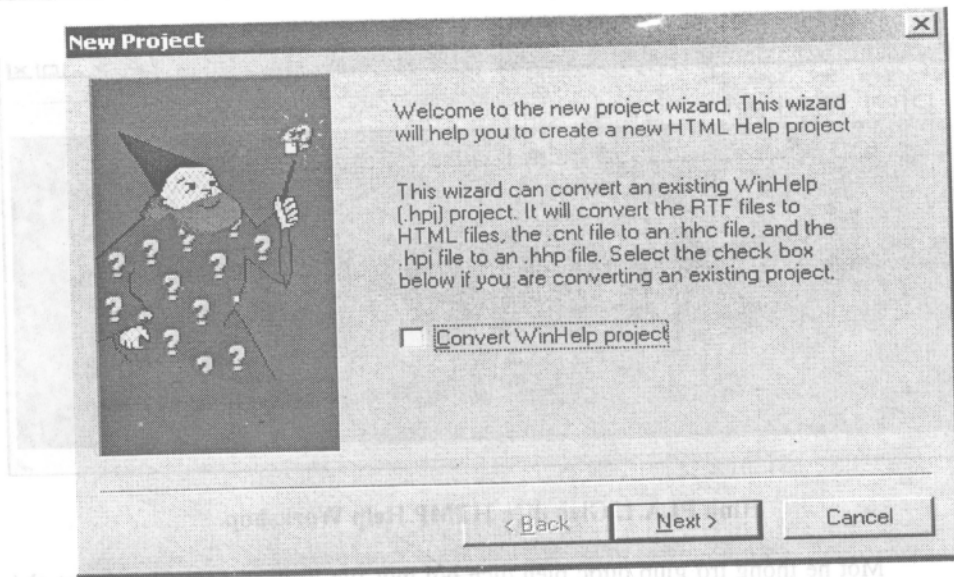
Một hệ thống trợ giúp được biên dịch bởi một file Project. Một file project chứa tất cả các trang chủ đề đã tạo ra của một hệ thống trợ giúp.

2. Để tạo một file Project từ thanh thực đơn: File → New → New dialog box xuất hiện



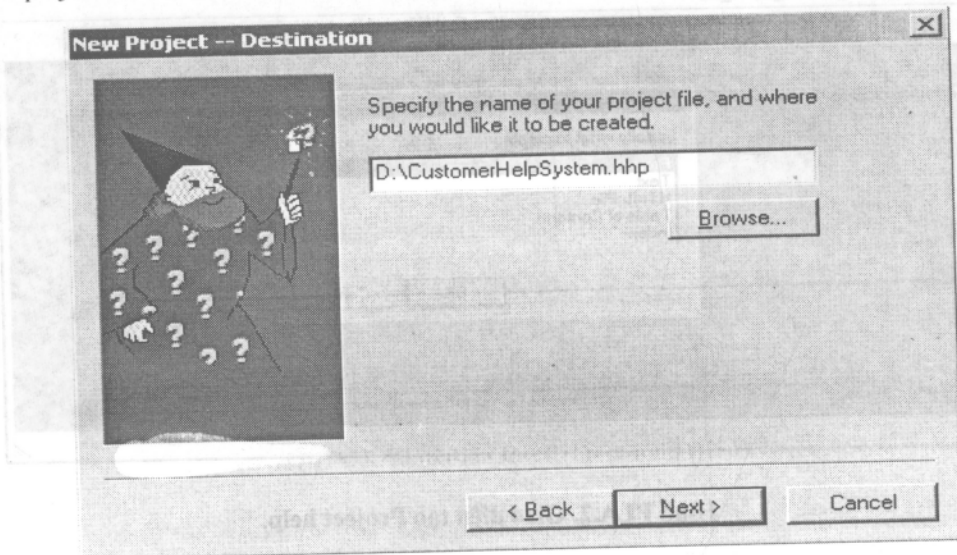
Hình PLA.2. Giao diện tạo Project help.

3. Lựa chọn Project được lựa chọn theo mặc định. Click OK.



Hình PLA.3. Giao diện tạo Project help (tiếp).

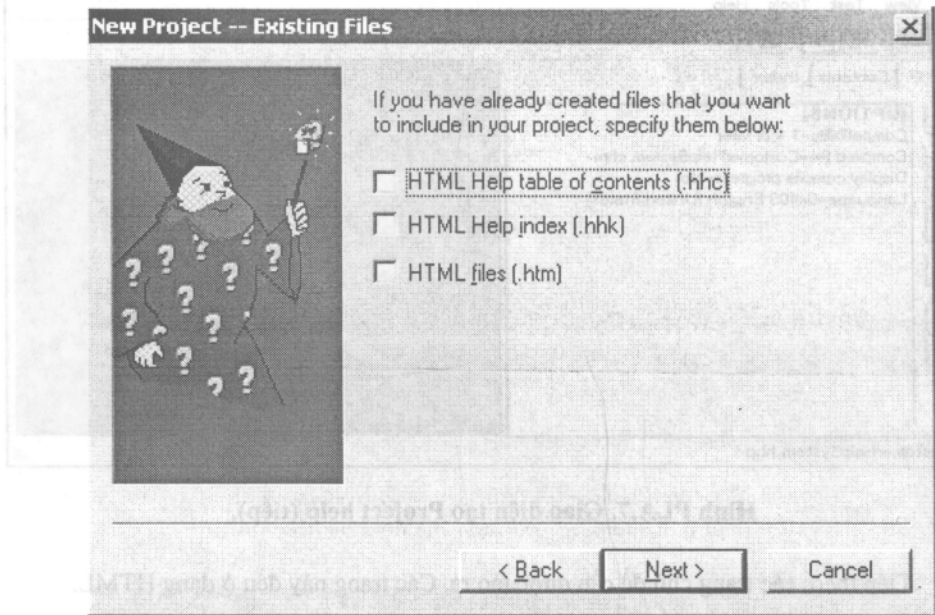
4. Click Next. Màn hình Project – Destination xuất hiện. Trên màn hình này, nhập tên cho project và đường dẫn bạn muốn tạo cho project.



Hình PLA.4. Giao diện tạo Project help (tiếp).

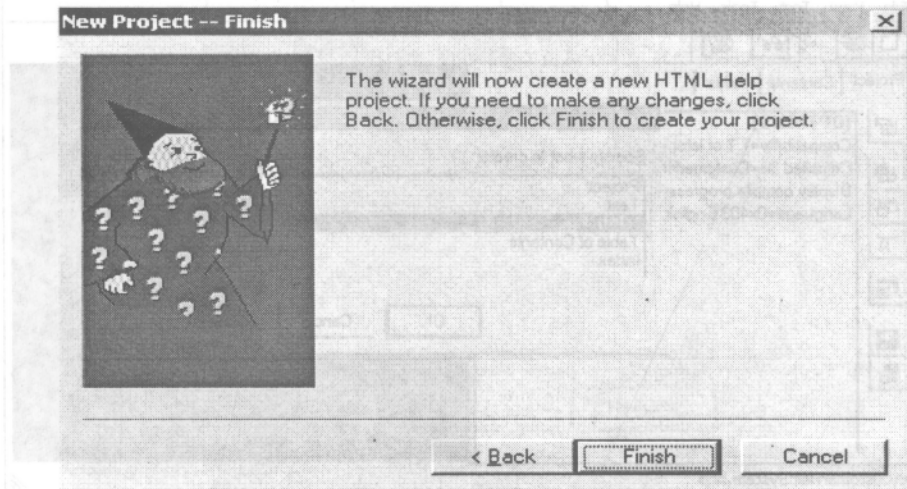
Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop

5. Click Next. Màn hình New Project – Existing Files xuất hiện. Nếu bạn có các file Help sẵn, như các file Index hoặc Content, bạn có thể thêm những file này vào trong file Project mà bạn đang tạo ra.



Hình PLA.5. Giao diện tạo Project help (tiếp).

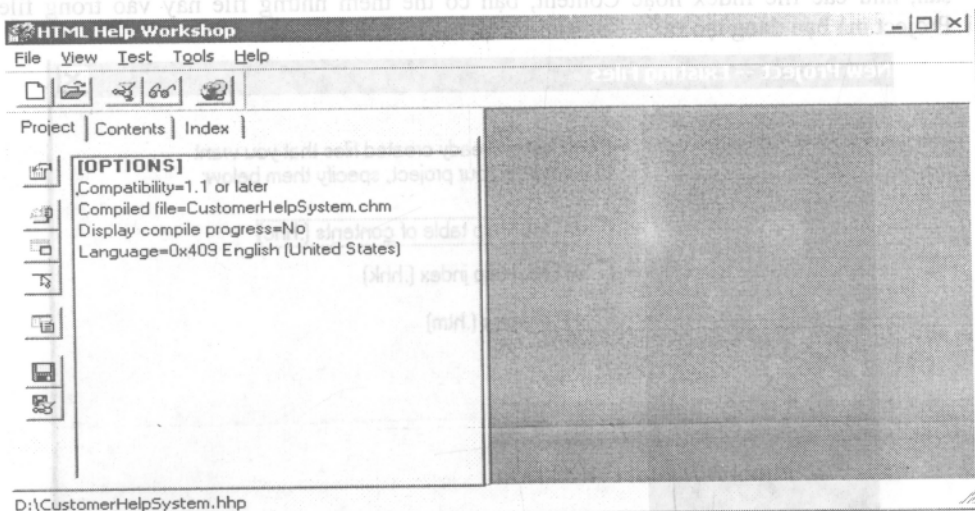
6. Màn hình New Project - Finish xuất hiện. Click Finish để tạo một file Project mới.



Hình PLA.6. Giao diện tạo Project help (tiếp).

Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop

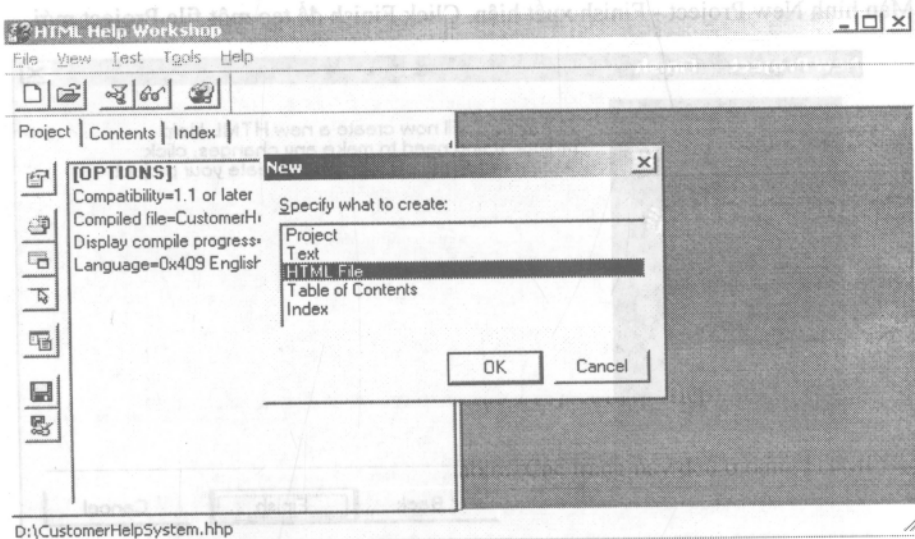
7. Màn hình sau sẽ xuất hiện. Tab Project được kích hoạt theo ngầm định.



Hình PLA.7. Giao diện tạo Project help (tiếp).

Tiếp theo, các trang chủ đề cần được tạo ra. Các trang này đều ở dạng HTML.

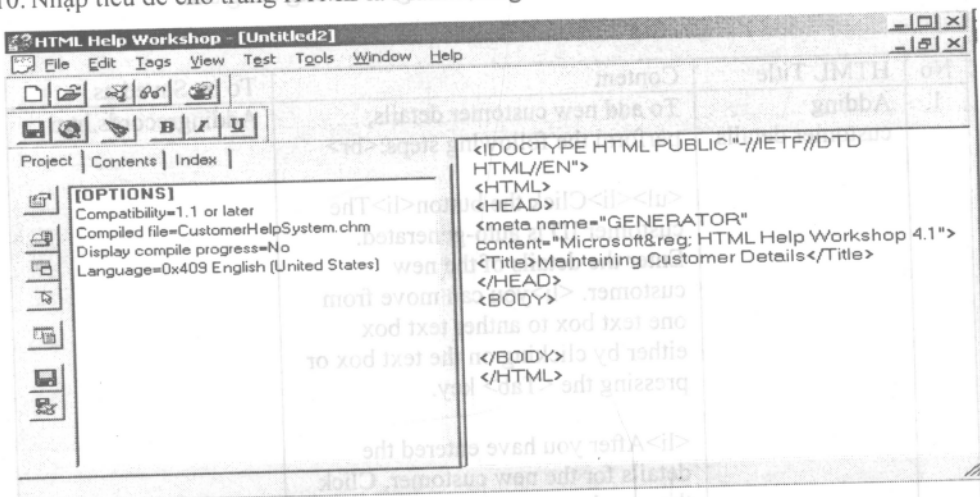
8. Để tạo một trang HTML, click File → New
9. Chọn HTML file từ hộp thoại New, click OK



Hình PLA.8. Giao diện tạo Project help (tiếp).

Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop

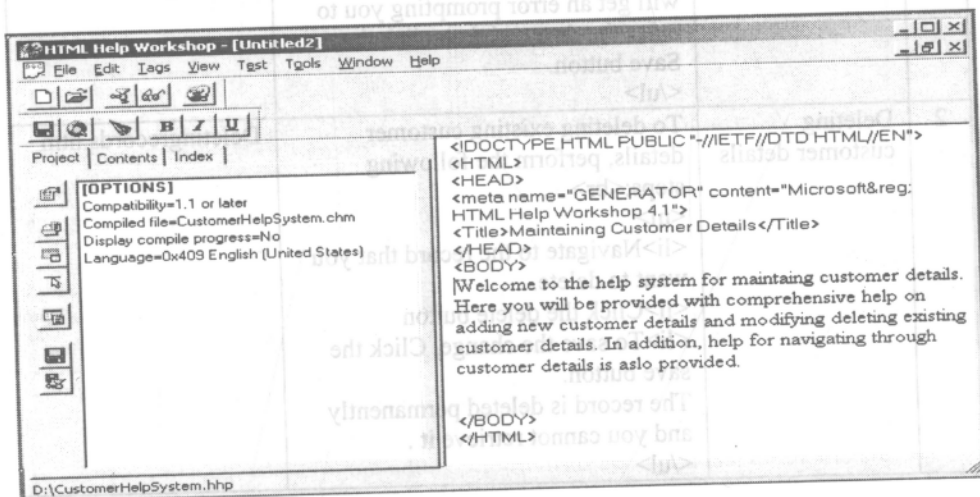
10. Nhập tiêu đề cho trang HTML là Maintaining customer details và click OK.



Hình PLA.9. Giao diện tạo tiêu đề trang HTML.

11. Nhập đoạn văn bản sau đây sau thẻ <Body> của file HTML trong cửa sổ bên tay phải và lưu tệp HTML với tên maintainingrecord.htm
"Welcome to the help system for maintaing customer details. Here you will be provided with comprehensive help on adding new customer details and modifying deleting existing customer details. In addition, help for navigating through customer details is aslo provided."

Trang HTML được chỉ ra trong hình PLA.10.



Hình PLA.10. Giao diện tạo trang HTML.

Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop

12. Tương tự, tạo ra 4 trang HTML với tên và nội dung như trong bảng sau:

No	HTML Title	Content	To be Saved as
1	Adding customer details	<p>To add new customer details, perform the following steps:
</p> <ul style="list-style-type: none">Click the buttonThe customer ID is auto-generated. Enter the details of the new customer. you can move from one text box to another text box either by clicking on the text box or pressing the <Tab> key. <p>After you have entered the details for the new customer, Click the save button to save the entered details.
</p> <p>You cannot leave the first name and the address of the customer blank.
</p> <p>Therefore, if you click the Save button with either the first name or the address text boxes blank, you will get an error prompting you to enter the data and then click the Save button.</p> <p></p>	Addingrecords.htm
2	Deleting customer details	<p>To deleting existing customer details, perform the following steps:
</p> <ul style="list-style-type: none">Navigate to the record that you want to delete.Click the delete buttonTo save the change, Click the save button. <p>The record is deleted permanently and you cannot retrieve it .</p> <p></p>	Deletingrecords.htm

Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop

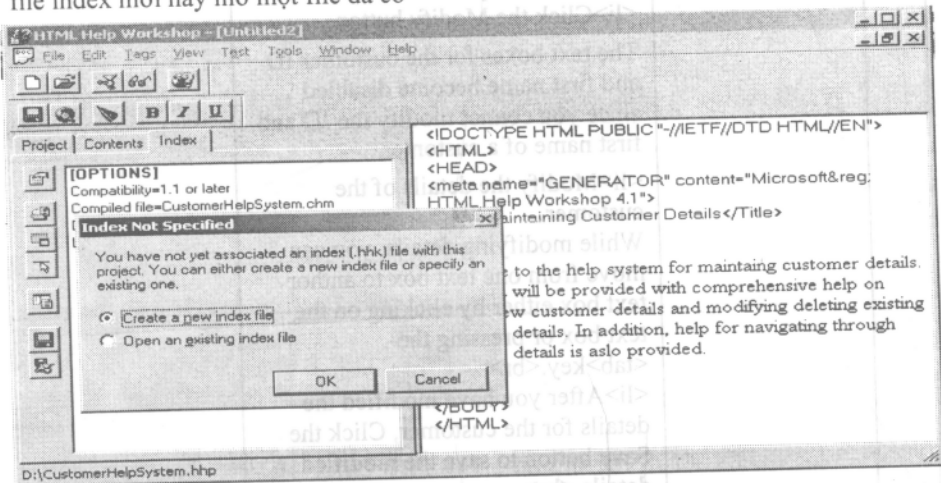
3	<p>Modifying Customer details</p>	<p>To modify existing customer details, perform the following steps:
Navigate to the record that you want to modify
Click the Modify button.
The text boxes for the customer ID and first name become disabled since you cannot modify the ID and first name of a customer.
Modify the details of the customer.
While modifying details, you can move from one text box to another text box either by clicking on the text box or pressing the <tab>key.
After you have modified the details for the customer, Click the Save button to save the modified details.
You cannot leave the first name and the address of the customer blank.

Therefore, if you click the save button with either the first name or the address text boxes blank, you will get an error prompting you to enter the data and then click the save button.</p> </p>	<p>Modifyingrecords.htm</p>
4	<p>Navigating through customer details</p>	<p>Use the following buttons to navigate through customer details:
First – Use this button to view the first customer record.
Next-Use this button to view the record next to the record that you are currently viewing.
Previous – Use this button to view the previous customer record.
Last – Use this button to view the last customer record.</p> </p>	<p>Navigatingrecords.htm</p>

Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop

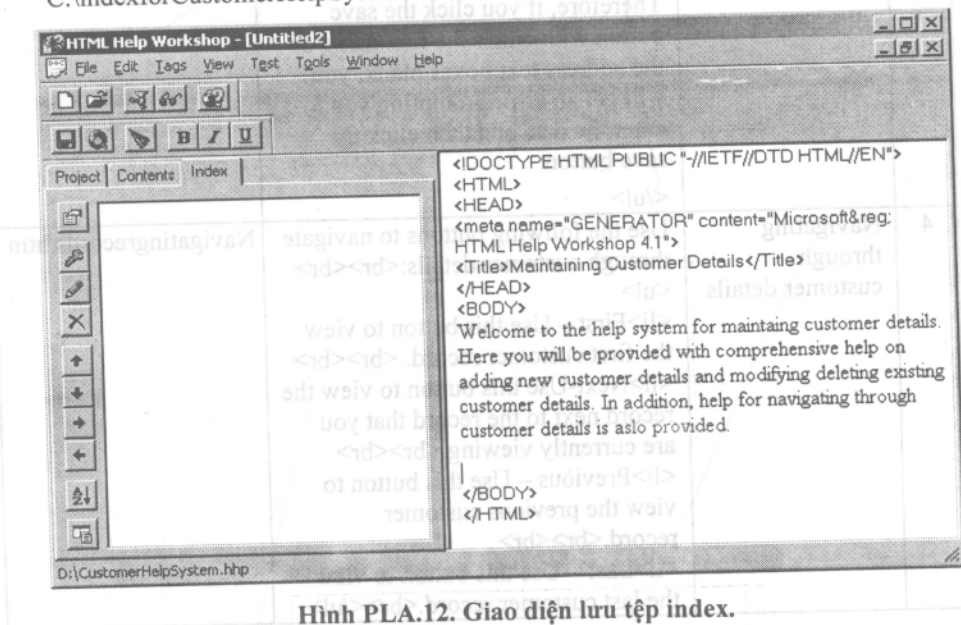
Sau khi các file HTML được tạo ra, bạn phải thêm vào các tính năng giúp người dùng xem một trang HTML bằng cách vào một từ khoá. Vì vậy, một file index cần được tạo ra cho các trang HTML.

- Để tạo ra một file index, click Index tab. Một message xuất hiện hỏi bạn muốn tạo một file index mới hay mở một file đã có



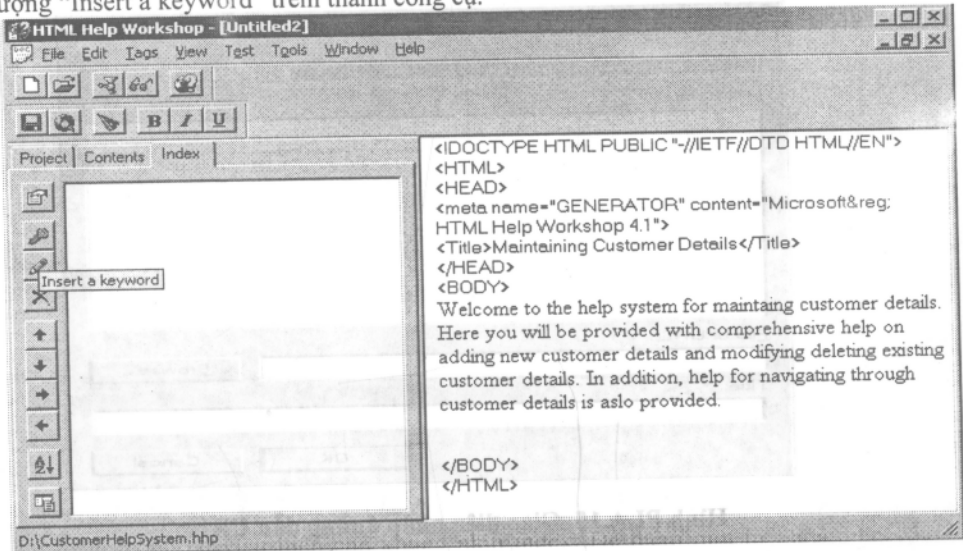
Hình PLA.11. Giao diện tạo index file.

- Mặc định, chọn "Create a new index file". Click OK.
- Trong hộp thoại Save, nhập tên và đường dẫn cho file index "C:\indexforCustomerHelpSystem" và click Save.



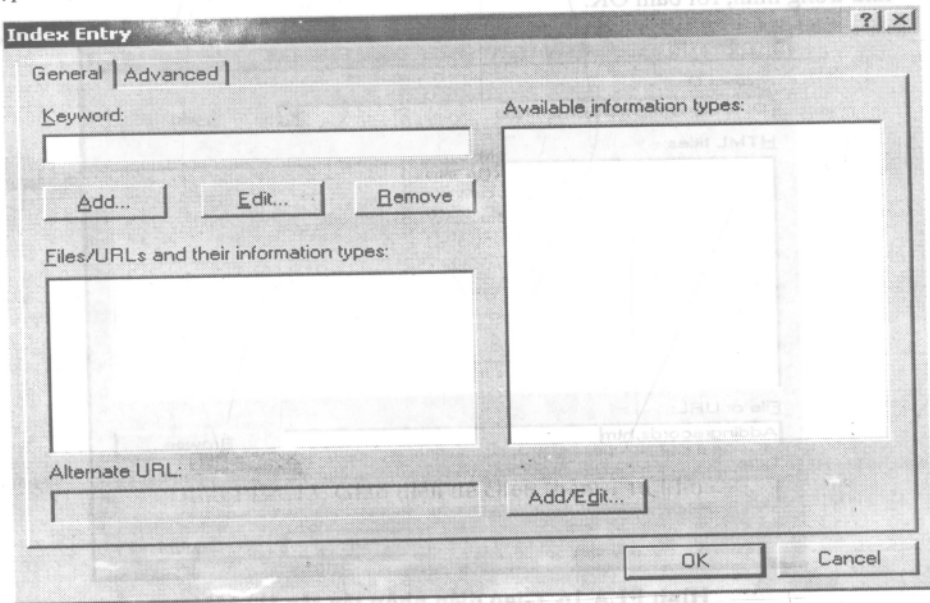
Hình PLA.12. Giao diện lưu tệp index.

16. Một thanh công cụ xuất hiện cho khung nhìn index. Để thêm một từ khoá, click biểu tượng “Insert a keyword” trên thanh công cụ.



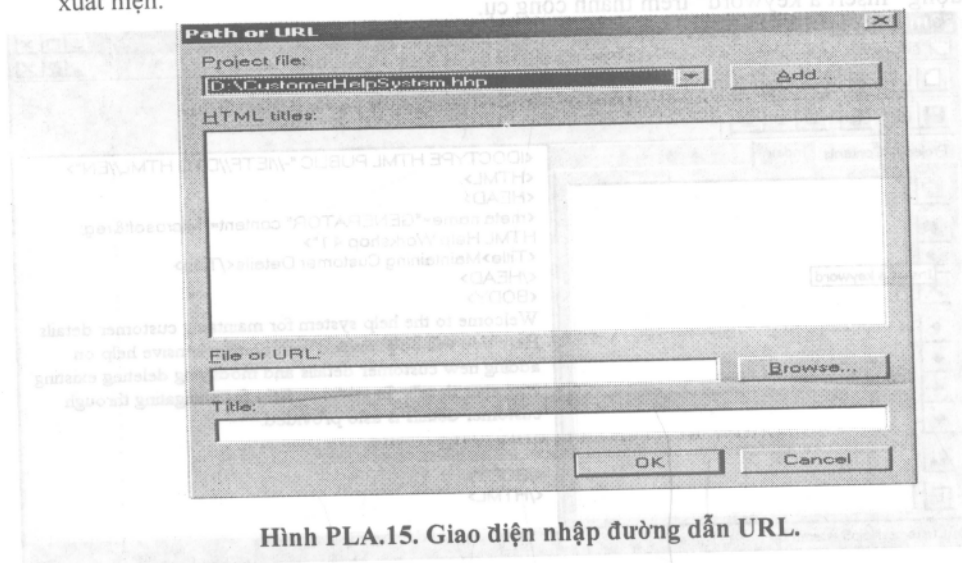
Hình PLA.13. Giao diện để chèn thêm 1 từ khoá.

17. Hộp thoại Index Entry xuất hiện



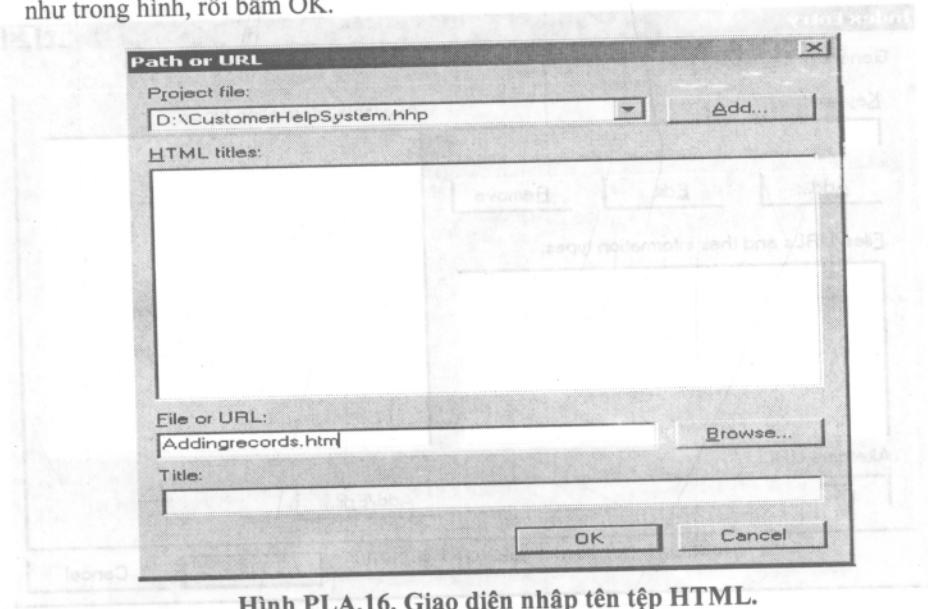
Hình PLA.14. Giao diện để chèn thêm 1 từ khoá (tiếp).

18. Nhập adding records trong hộp văn bản Keyword và click Add. Hộp thoại Path or URL xuất hiện.



Hình PLA.15. Giao diện nhập đường dẫn URL.

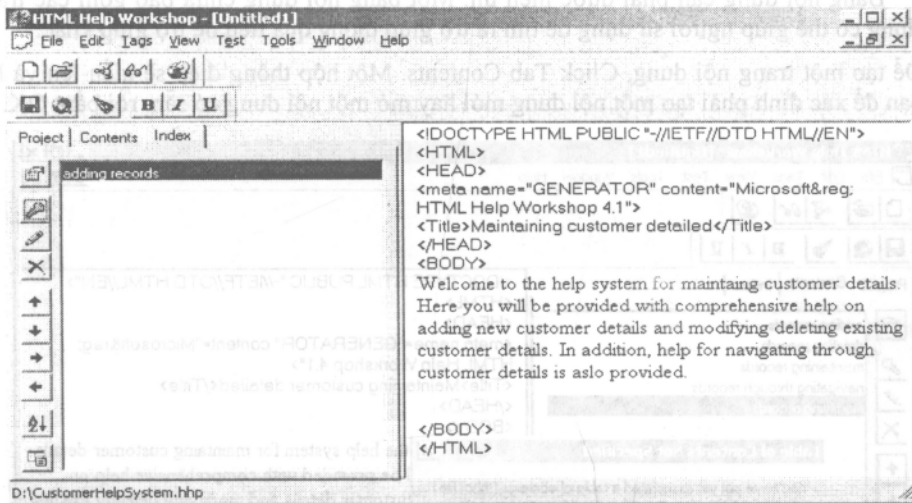
19. Bằng mặc định, Project đang mở được hiển thị trong list box của file project. Sau đó nhập tên file HTML kết hợp với từ khoá bởi enter tên file trong text File or URL hoặc là chọn lấy file bằng cách bấm nút Browse. Enter tên file name là addingrecords.htm như trong hình, rồi bấm OK.



Hình PLA.16. Giao diện nhập tên tệp HTML.

Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop

Click OK, màn ảnh hiển thị như trong hình PLA.15.

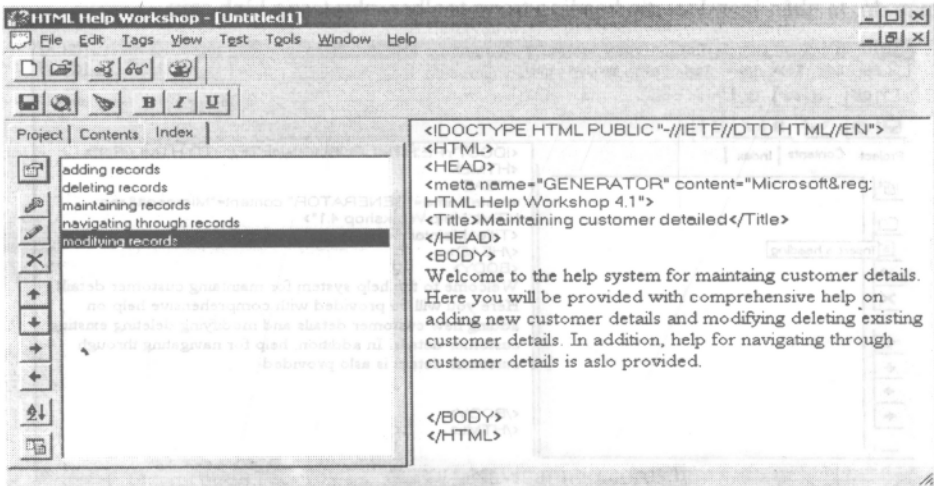


Hình PLA.17. Giao diện toàn trang HTML và tệp lưu trữ.

20. Tương tự, ta chèn vào những từ khóa sau:

Keyword	File to linked with the keyword
Deleting record	Deletingrecords.htm
Maintaing records	Maintaingrecords.htm
Navigating through records	Navigatingrecords.htm
Modifying records	Modifyingrecords.htm

Màn hình có dạng:

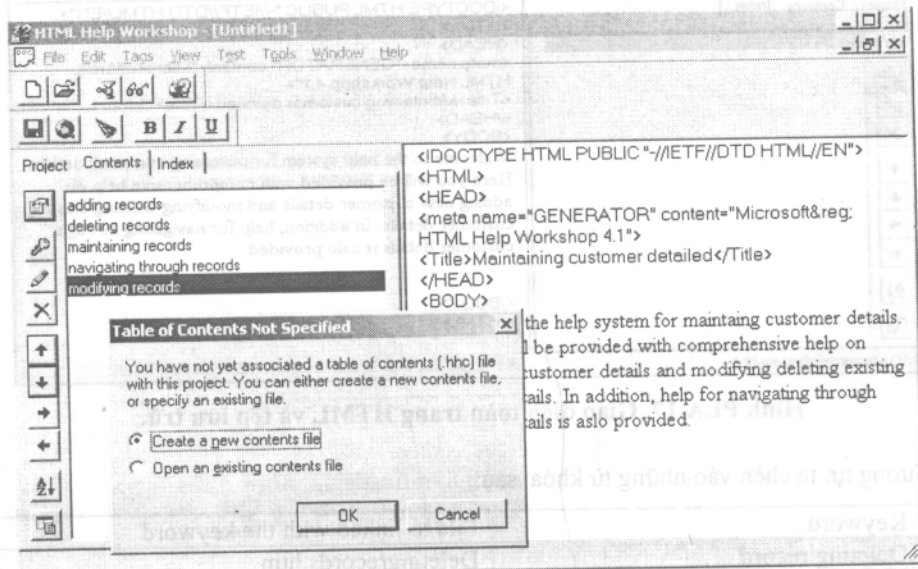


Hình PLA.18. Giao diện toàn trang HTML sau khi bổ sung các từ khóa.

Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop

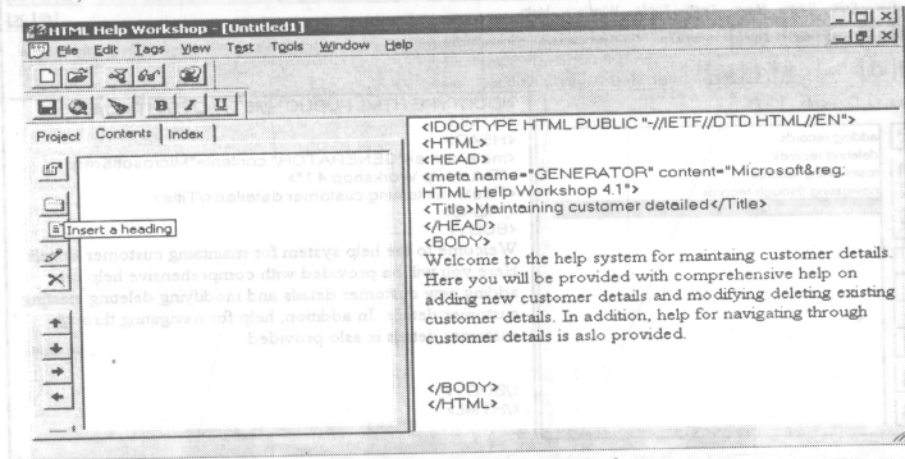
Bảng nội dung cần phải được hiển thị. Một bảng nội dung chứa bao gồm các trang nội dung có thể giúp người sử dụng để tìm ra trợ giúp thông qua tiêu đề trợ giúp khác.

- Để tạo một trang nội dung, Click Tab Contents. Một hộp thông điệp sẽ hiển lên và hỏi bạn để xác định phải tạo một nội dung mới hay mở một nội dung có sẵn, rồi bấm OK.



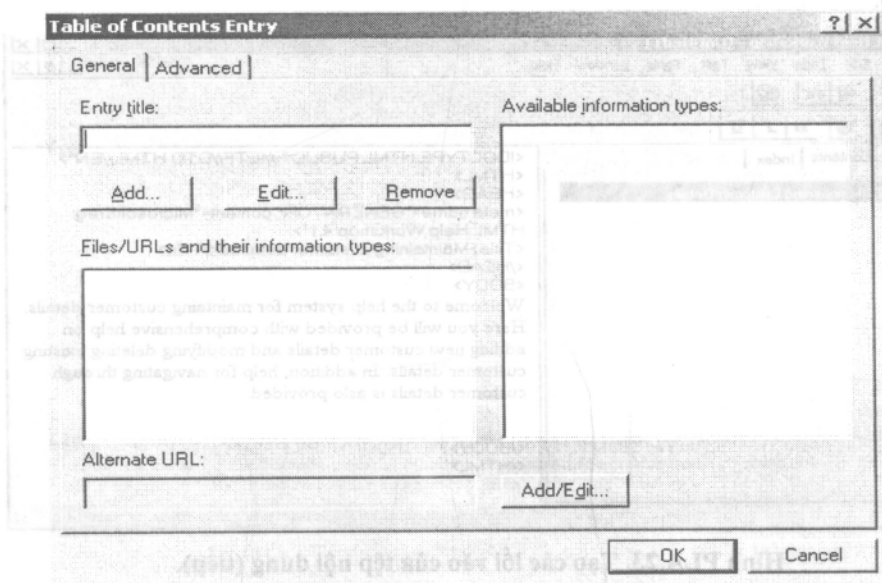
Hình PLA.19. Giao diện tạo trang nội dung.

- Nhập đường dẫn và tên của bảng file nội dung là D:\tocforCustomerHelpSystem và lưu lại.
- Để tạo một heading của file nội dung phục vụ cho việc tìm kiếm của người sử dụng nào đó, ta nhấn icon Insert a heading trong toolbar, như trong hình sau:



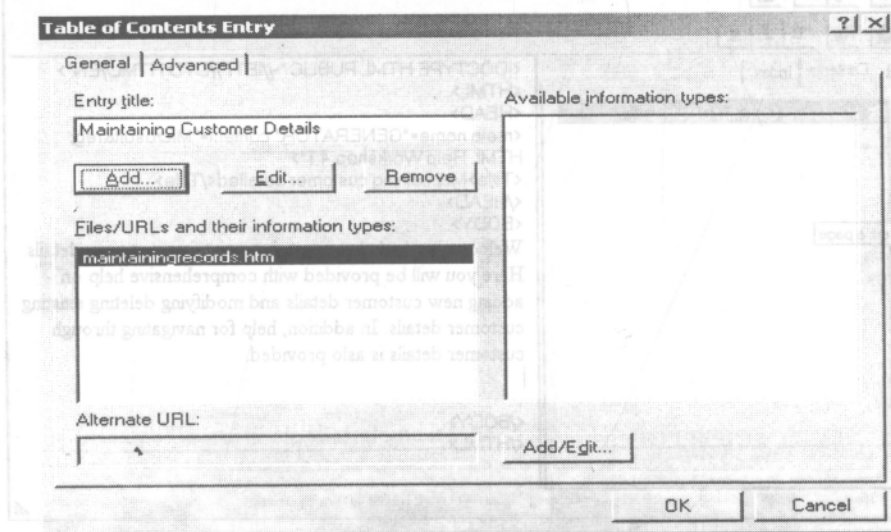
Hình PLA.20. Giao diện chèn tiêu đề trang.

24. Hộp thoại Table Of Contents Entry sẽ xuất hiện như trong hình PLA.19.



Hình PLA.21. Tạo các lối vào của tệp nội dung.

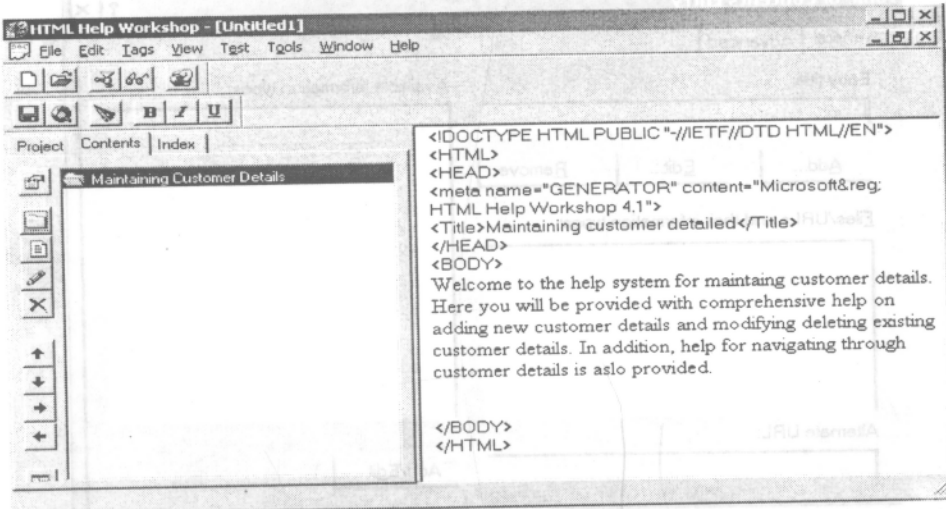
25. Nhập từ "Maintaing Customer Details" trong hộp văn bản Entry title và bấm nút Add
26. Enter maintainingrecords.htm trong text box File Or URL và Click OK.
27. Hộp thoại Table Of Contents Entry sẽ hiển thị như hình PLA.20.



Hình PLA.22. Tạo các lối vào của tệp nội dung (tiếp).

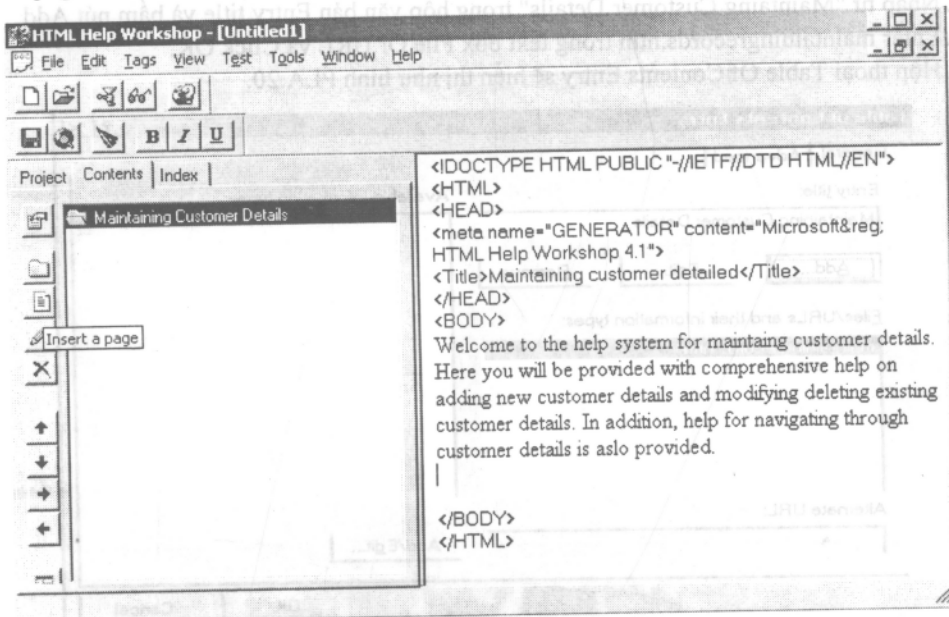
Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop

Cửa sổ sẽ thấy như hình sau. Heading mà bạn đã tạo ra được hiển thị có hình icon folder trước nó.



Hình PLA.23. Tạo các lỗi vào của tệp nội dung (tiếp).

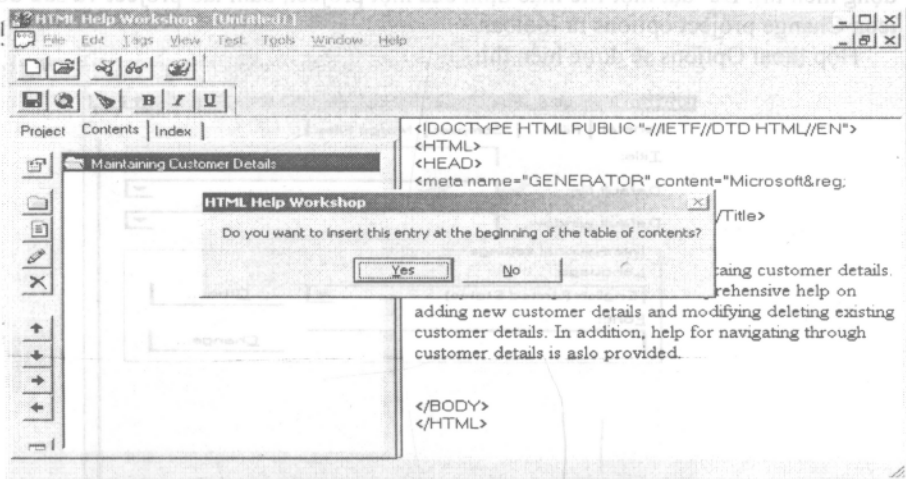
28. Sau khi heading được tạo, nội dung của các trang phải được nhập vào. Nhấn icon Insert a page trên toolbar.



Hình PLA.24. Chèn thêm trang.

Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop

29. Hộp thông điệp xuất hiện sẽ hỏi mình trang này phải được insert vào đầu tiên của bảng file nội dung hay không?



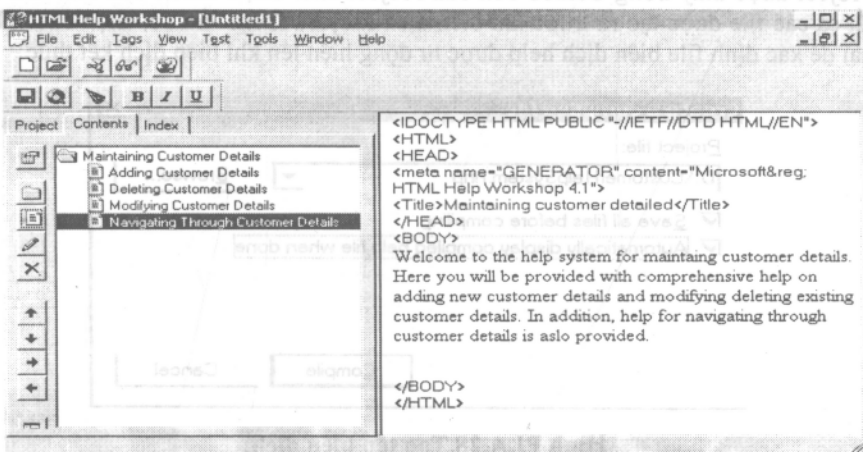
Hình PLA.25.Chèn thêm trang (tiếp).

30. Nhấn No. Trong hộp thoại Table of Contents Entry và đánh từ khoá Adding Customer Details trong text box Entry title và enter addingrecords.htm trong text box File or URL.

31. Tương tự thêm vào trang nội dung sau:

Entry Title	File to linked with the title
Deleting Customer Details	Deletingrecords.htm
Modifying Customer Details	Modifyingrecords.htm
Navigating Through Customer Details	Navigatingrecords.htm

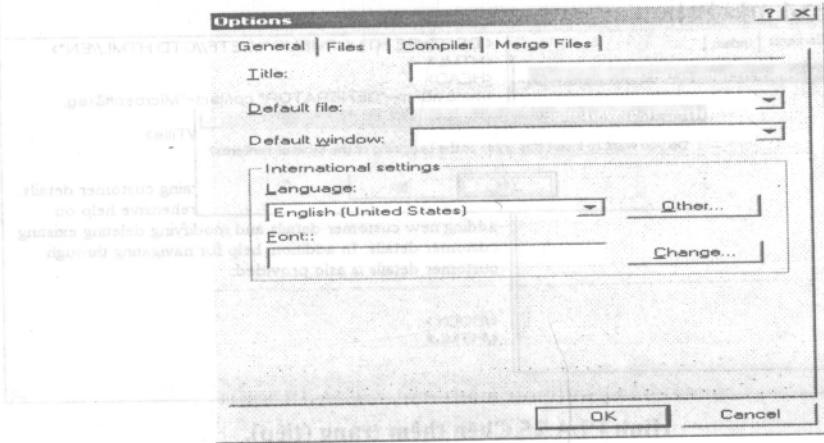
Cửa sổ sẽ thấy như sau:



Hình PLA.26.Chèn thêm nội dung vào trang.

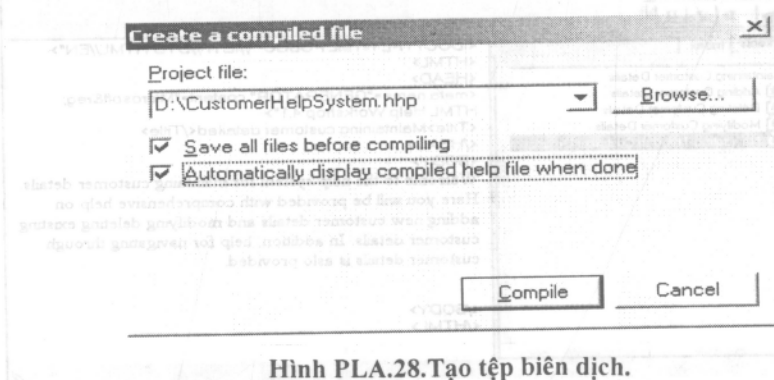
32. Bạn có thể chỉ định một file HTML mặc định của project bạn. Khi file help được hiển thị của một người sử dụng nào đó, file HTML đặt mặc định của một Project được mở và tự động hiển thị. Để đặt một file mặc định của một project, bấm tab project và sau đó bấm icon Change project options từ toolbar.

Hộp thoại Options sẽ được hiển thị:



Hình PLA.27. Tạo hộp hội thoại.

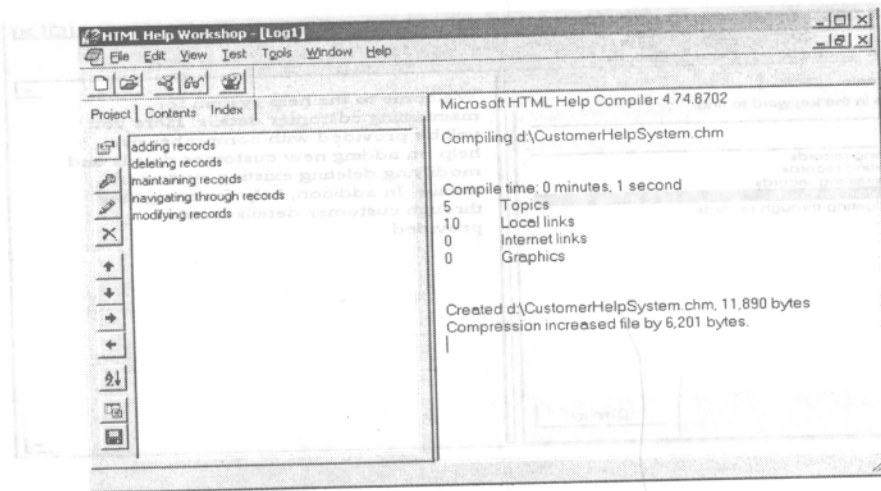
33. Enter file maintainingrecords.htm là file mặc định trong Combo box Default File và bấm OK.
34. Sau khi file index và bảng nội dung được thêm vào trong Project, bạn cần dịch Project file để insert và save tất cả các files trong folder Project. Để biên dịch Project, bấm vào File từ thanh menu và chọn Compile từ file menu.
35. Hộp thoại Create a compiled File sẽ hiển thị. Bằng mặc định, đường dẫn hiện tại của file Project được thấy trong Combo box của Project file. Chọn check box đầu tiên để save tất cả các file được tạo ra trước khi biên dịch Project. Thêm vào đó, chọn check box thứ hai để xác định file biên dịch help được tự động hiện lên khi biên dịch kết thúc.



Hình PLA.28. Tạo tệp biên dịch.

Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop

36. Click nút Compile để bắt đầu biên dịch file project. File Project được biên dịch là file .chm. Sau khi biên dịch xong, thông tin biên dịch được thấy như hình sau:

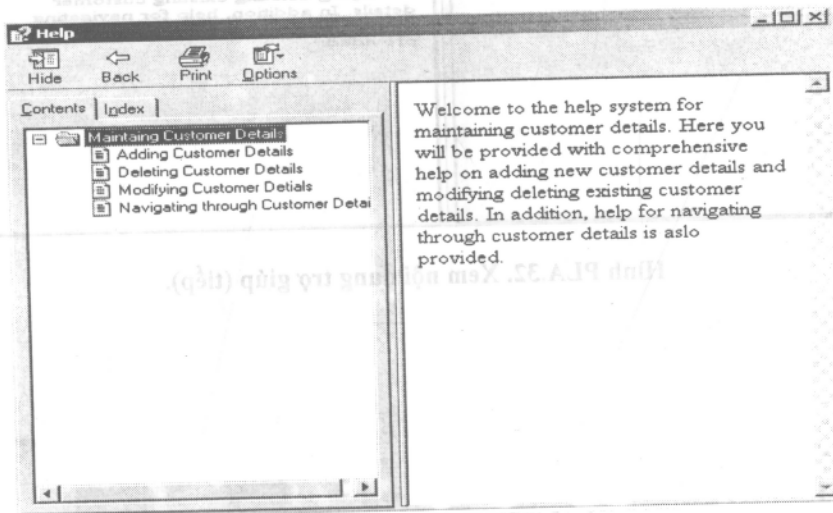


Hình PLA.29. Cửa sổ đề án đã biên dịch.

4. Truy nhập hệ thống trợ giúp

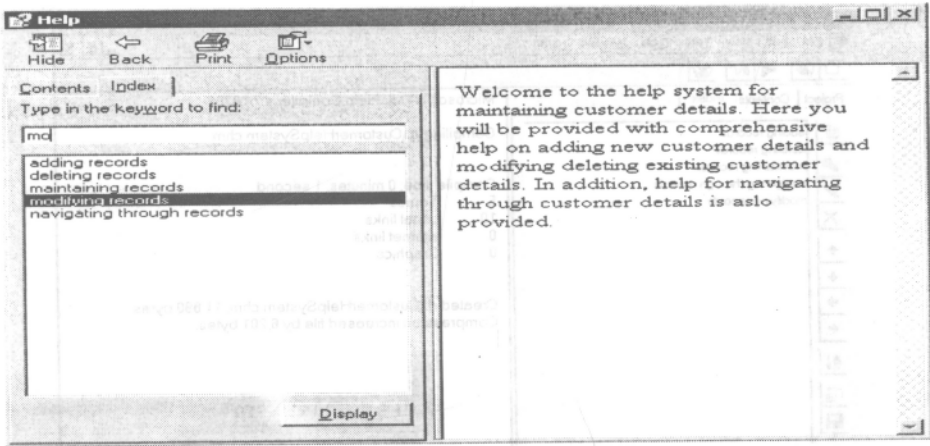
Hành động:

1. Để xem file help đã biên dịch, bạn click vào View từ thanh menu. Sau đó chọn Compile File từ menu View. Enter đường dẫn của file help trong Combo box của Compiled File.
2. Sau khi bấm View rồi nó sẽ hiển ra như hình sau:



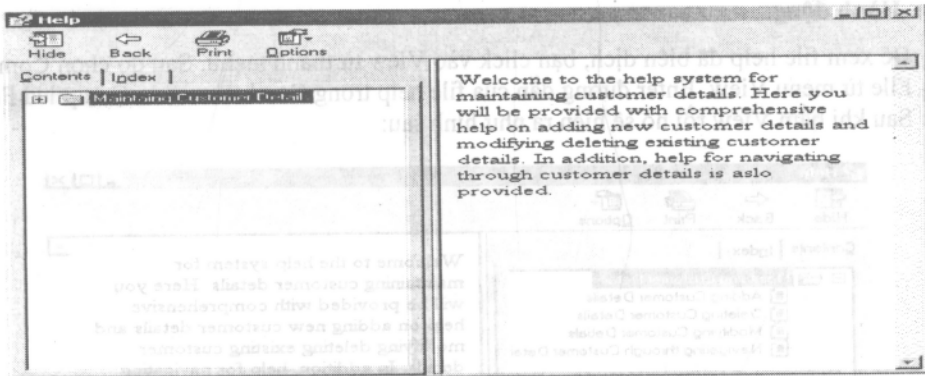
Hình PLA.30. Nội dung tệp trợ giúp đã biên dịch.

Khi người sử dụng bắt đầu đánh vào một từ khoá nào đó, nó sẽ tự động tìm đến chủ đề đó rồi bạn có thể Double-Click của chủ đề ấy để xem nội dung.



Hình PLA.31. Xem nội dung trợ giúp.

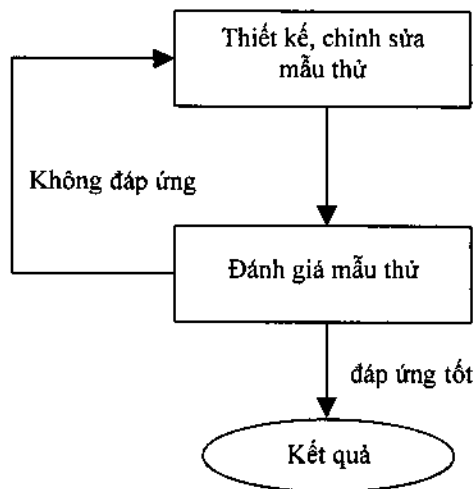
Khi bạn bấm trên tab Contents bạn sẽ thấy như trong hình sau:



Hình PLA.32. Xem nội dung trợ giúp (tiếp).

Phụ lục B
CÁC CÔNG CỤ TẠO LẬP MẪU THƯ
HYPERCARD VÀ REVOLUTION

Trong thiết kế giao diện, việc sử dụng mẫu thư là cần thiết. Ở đó, người ta đưa ra các ý kiến của mình thông qua việc trình diễn của các mẫu thư, các yêu cầu người dùng về giao diện qua đó được xác định lại. Quy trình xây dựng mẫu thư được chỉ ra như trong hình PLB.1.



Hình PLB.1. Mô hình mẫu thư giao diện

Có khá nhiều công cụ xây dựng mẫu thư, song trong phụ lục này, chúng tôi giới thiệu hai công cụ khá quen thuộc đó là: HyperCard và Revolution.

A. HYPERCARD

1. Giới thiệu về HyperCard

HyperCard là một gói các công cụ phần mềm được viết để tạo ra các ứng dụng tương tác đa phương tiện trên môi trường MacOS. Sử dụng HyperCard có thể tạo ra nhiều ứng dụng khác nhau bao gồm cả hoạt hình, xử lý văn bản, cơ sở dữ liệu và các thành phần bảng tính.

Phụ lục B: Công cụ tạo lập mẫu thử

HyperCard bao gồm các stack, và mỗi stack bao gồm các card, các card trong một stack có thể sắp xếp theo trật tự khác nhau. Cũng như vậy, việc thêm vào hay xóa đi các stack (có chứa hình hoạt hình hay âm thanh...) là khá dễ dàng.

Mỗi Card bao gồm:

- Graphics: các đường thẳng, các hình khối, clip-art, và công cụ vẽ bằng tay.
- Các trường văn bản.
- Các nút (có thể dùng để kết nối các Card hoặc xử lý các sự kiện).

2. Các tính năng của HyperCard

- Hỗ trợ ngôn ngữ siêu văn bản (Hyper Text) và cũng sử dụng cấu trúc lập trình khi thiết kế Web, HTML và Javascript.
- Thiết kế giao diện một cách dễ dàng, các Card có thể kết nối với nhau.
- Hỗ trợ Multimedia với các công cụ trình diễn đa phương tiện với các chức năng giống PowerPoint. Nó có thể tạo ra các trình diễn đa phương tiện như hoạt hình, hành động, âm thanh, hình ảnh....
- Khởi chạy và điều khiển các chương trình khác sử dụng lệnh HyperTalk đơn bắt vào các sự kiện của nút để khởi chạy (Các Stack trong HyperCard cho phép mở các ứng dụng và các tệp bất kỳ nơi nào trong máy tính).
- Hỗ trợ giao tiếp trong mạng (mẫu thử trình ứng dụng nhóm..).

1. Ngôn ngữ HyperTalk trong Hypercard

Hypercard cũng như Revolution (để cập trong phần sau) đều dùng ngôn ngữ HyperTalk để viết các câu lệnh cho phần trình diễn của các card. HyperTalk mang các đặc điểm của một ngôn ngữ script (không linh động như các ngôn ngữ chuyên nghiệp C/Java...) và mang tính tương tác cao với người dùng. Các mẫu thử trong HyperCard là các điều khiển (control), do đó khi tạo tương tác với các điều khiển này, ta phải làm việc với các sự kiện gây tương tác.

Các sự kiện của các điều khiển được dùng chủ yếu khi tạo mẫu thử với HyperTalk:

- mouseUp: xảy ra khi chuột nhấn xuống một điều khiển.
- mouseDown: xảy ra khi chuột được thả ra sau sự kiện nhấn vào một điều khiển.
- mouseClick: xảy ra khi người dùng click chuột vào điều khiển.
- keyDown: xảy ra khi người dùng nhấn phím.
- keyUp: xảy ra khi người dùng thả phím.
- preOpenStack: xảy ra khi stack có điều khiển đang gắn vào được mở ra. Các lệnh trong sự kiện này sẽ được thực hiện trước khi mở stack ra.
- OpenStack: xảy ra khi stack có điều khiển đang gắn vào được mở ra. Các lệnh trong sự kiện này sẽ được thực hiện sau khi mở stack ra.
- preCloseStack: xảy ra khi stack có đang gắn vào bị đóng lại. Các lệnh trong sự kiện này sẽ được thực hiện trước khi stack bị đóng.
- CloseStack: xảy ra khi stack có điều khiển đang gắn vào bị đóng lại. Các lệnh trong sự kiện này sẽ được thực hiện sau khi stack bị đóng.

Phụ lục B: Công cụ tạo lập mẫu thử

- preOpenCard, OpenCard, preCloseCard, CloseCard: các sự kiện này, như tên gọi của nó, cũng tương tự như với stack.

Một số lệnh trình diễn điển hình:

1. Lệnh show

Cú pháp:

```
show object [with visual [effect] effectName]
```

Ví dụ:

```
show stack "Help Palette"  
show image "Splash" with visual effect dissolve
```

Giải thích:

Lệnh show dùng để hiện một đối tượng đang bị ẩn

- object là tên của đối tượng;
- effectName là tên của hiệu ứng.

Như ở ví dụ trên, dòng 1 sẽ hiển thị stack có tên "Help Palette". Ở dòng sau, lệnh này sẽ hiện ảnh đã được định nghĩa với tên Splash với hiệu ứng dissolve (tan đi).

2. Lệnh hide

Cú pháp:

```
hide object [with visual [effect] effectName]
```

Ví dụ:

```
hide field "New" with visual effect dissolve  
hide this card with visual effect plain
```

Giải thích:

Lệnh hide dùng để ẩn một đối tượng đang hiển thị

- object là tên của đối tượng;
- effectName là tên của hiệu ứng.

3. Lệnh open stack

Cú pháp

```
open stack "stackName"
```

Ví dụ:

```
open stack "ViewDoc"
```

Giải thích:

Lệnh open stack dùng để mở một stack, hoặc substack đang bị ẩn có tên stackName. Như ở ví dụ trên, dòng lệnh này sẽ mở một stack có tên "ViewDoc".

4. Lệnh close stack

Cú pháp:

```
close stack "stackName"
```

Ví dụ:

```
close stack "ViewDoc"
```

5. Lệnh go

Cú pháp:

```
go to card [of stack]
go to next [marked] card
go next
go back
go to stack stackName
```

Một số lệnh rẽ nhánh:

1. Lệnh if... then

Cú pháp:

```
if <expression> then
    --các lệnh 1
[else
    -- các lệnh 2]
end if
```

Ví dụ:

Trong một đoạn bắt sự kiện mouseUp của một trường:

```
on mouseUp
    if me is "Tokyo" then
        put "Đây là thủ đô của Nhật" into
field "Ten thu do"
    else
        put "Không biết thủ đô này!" into
field "Ten thu do"
    end if
end mouseUp
```

Giải thích: Đây là lệnh rẽ nhánh có điều kiện

2. Lệnh Repeat

Cú pháp:

```
repeat <expression>
    -- các lệnh
end repeat
```

Ví dụ:

```
repeat for 3 times
    beep
end repeat
repeat with x=1 to 3
    beep
end repeat
repeat until the mouseClicked
    go next
    wait for 100 millisecond
end repeat
```

3. Lệnh switch

Phụ lục B: Công cụ tạo lập mẫu thử

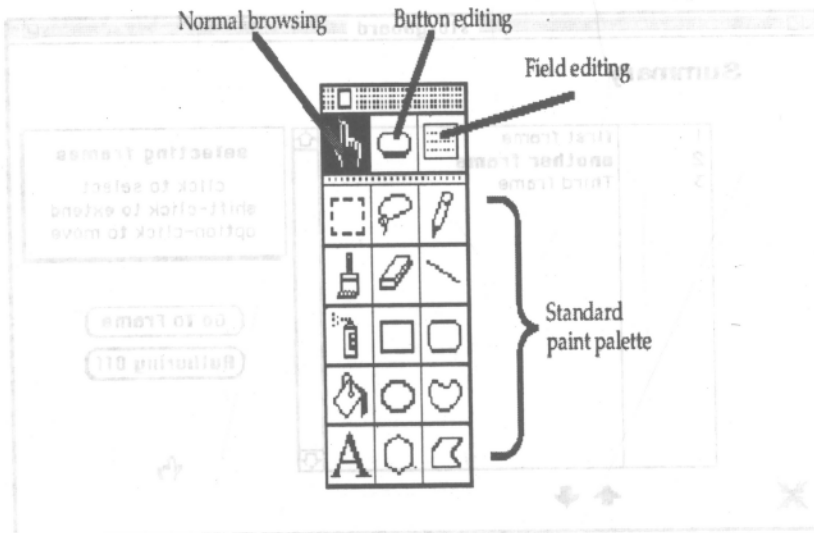
Cú pháp:

```
switch <expression>
  case <one>
    --các lệnh 1
  break
  case <two>
    --Các lệnh 2
  break
end switch
```

Ví dụ:

```
on menuPick pWich
  switch pWhich
  case "New"
    open stack "NewDoc"
  break
  case "Open"
    open stack "OpenDoc"
  break
  case "Exit"
    close this stack
  break
end switch
end menuPick
```

3. Làm việc với HyperCard (example)



Hình PLB.2. Tool Palette của HyperCard.

3.1. Sử dụng background

Mỗi thẻ đều ở trong một background và background chứa những nội dung sau:

- Các biểu tượng, các mảng và các nút.
- Được chia sẻ bởi tất cả các thẻ.
- Cũng có những nút thẻ riêng biệt.
- Những thành phần của thẻ luôn xuất hiện ở đầu.

3.2. Sắp xếp các background

- Sử dụng **B** để sắp xếp các background.
- Xoá bỏ các thành phần của thẻ.
- Làm tiếp như là các thẻ riêng biệt.
- Tạo một bảng Object cho background mới.

3.2. Một storyboard đơn giản

Các trường đặc biệt

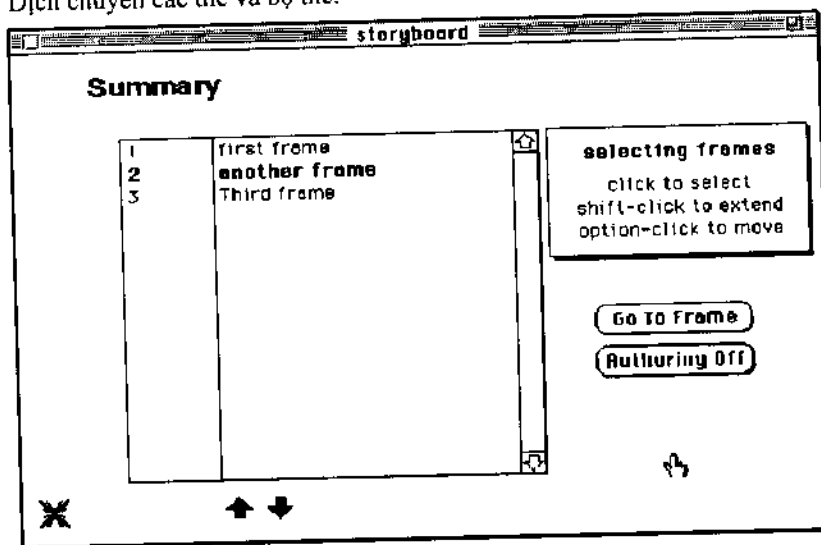
- Tên màn hình.
- Số trình tự.

Bảng điều khiển

- Di chuyển các thẻ.
- Ẩn hiện tên.

3.4. Thẻ tóm tắt

- Thể hiện cái nhìn tổng quan về trình bày.
- Bỏ qua những điểm đặc biệt.
- Dịch chuyển các thẻ và bộ thẻ.

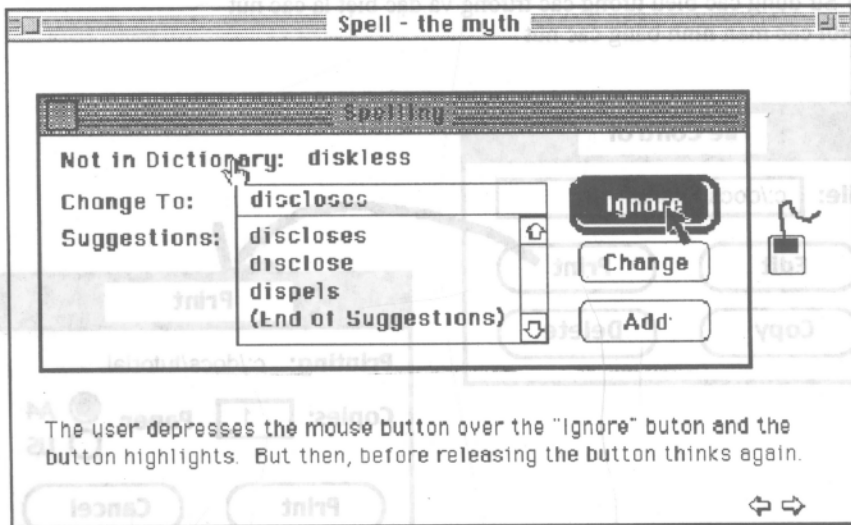


Hình PLB.3. Một storyboard đơn giản.

Phụ lục B: Công cụ tạo lập mẫu thử

3.5. Extra Flourishes

- Sử dụng các link cho đường dẫn thay thế.
- Thêm những đoạn có giải thích.



Hình PLB.4. Người dùng có thể thêm chú thích.

3.6. Đoạn hội thoại tuần tự

- Đó là mô hình của sự tương tác.
 - Một số biến đổi không mang nhiều ý nghĩa.
 - Một số thì có.
 - Những lệnh thay thế.
- Các đường dẫn thay thế

3.7. Storyboard

- Có một sequence cố định.
- Dialogue
- Tắt các các trạng thái.
 - Tắt cả các đường dẫn.
 - Tắt cả những cái để thay thế.

3.8. Sử dụng HyperCard

- Mỗi card là một trạng thái.
- Các nút được dịch chuyển giữa các trạng thái.

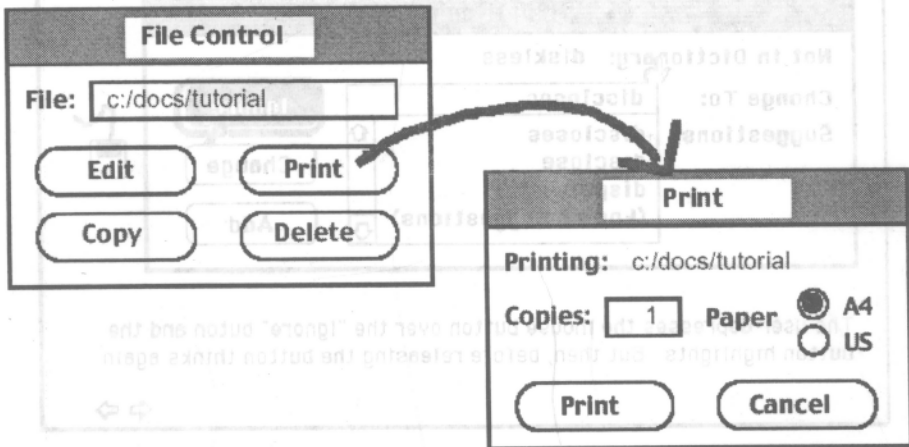
3.9. Xây dựng đối thoại

- Đặc điểm
- Phác thảo các trạng thái chính.

Phụ lục B: Công cụ tạo lập mẫu thư

Tạo ra các thẻ cho mỗi trạng thái

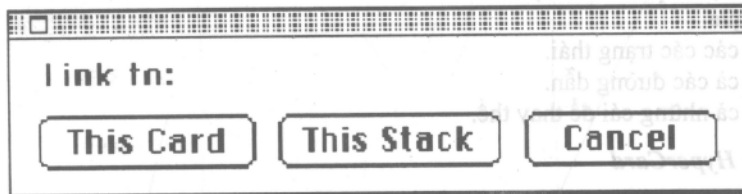
- Sao chép các thẻ
 - Sử dụng các background
 - Sử dụng các biểu tượng các trường và đặc biệt là các nút
- Nổi các màn hình bằng các nút



Hình PLB.5. Xây dựng giao diện điều khiển in.

3.10. Tạo link

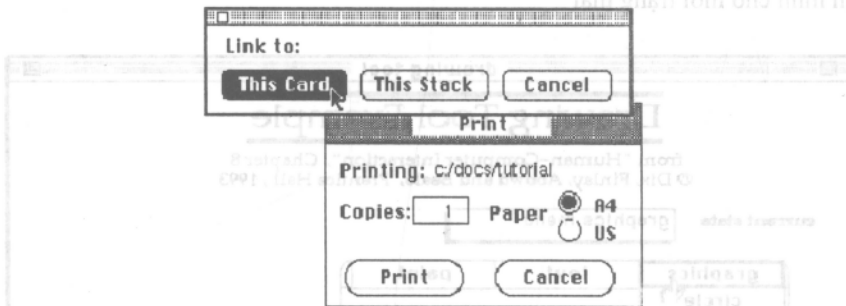
- Chọn bộ các nút
 - Chọn nút để nổi
 - Sử dụng 'Button info...' từ 'Objects'
- Nhấn vào 'Link to...'



Hình PLB.6. Giao diện tạo Link.

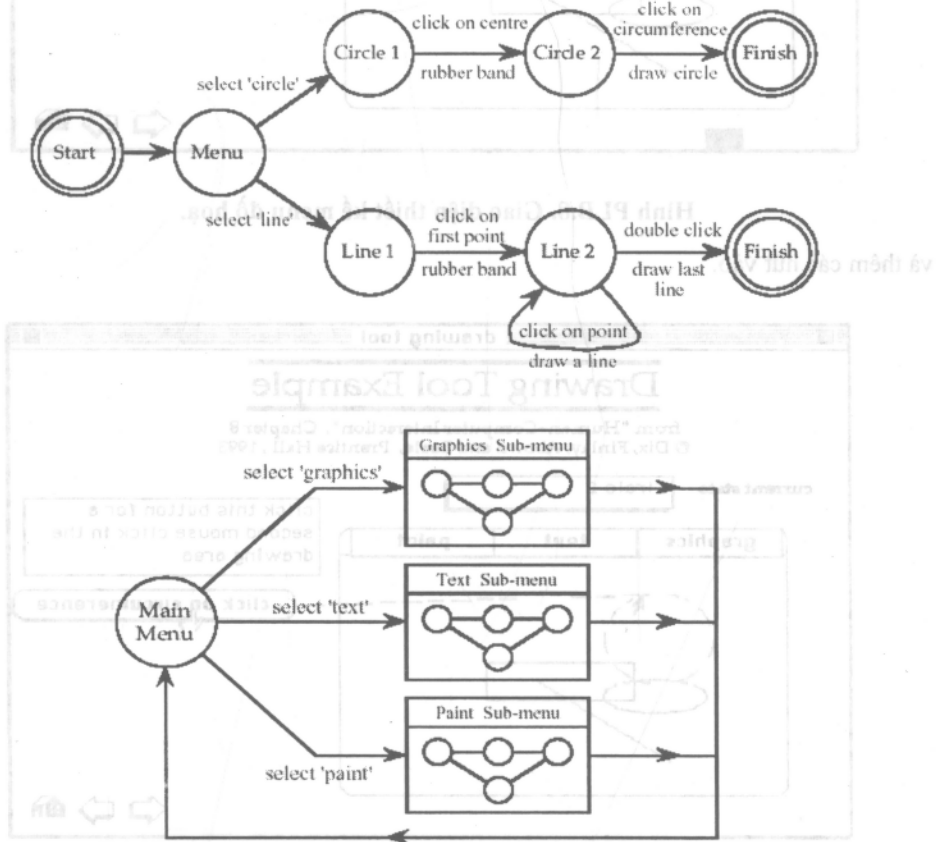
- + Chuyển đến thẻ nhận
- Ấn vào 'This card'

Phụ lục B: Công cụ tạo lập mẫu thử



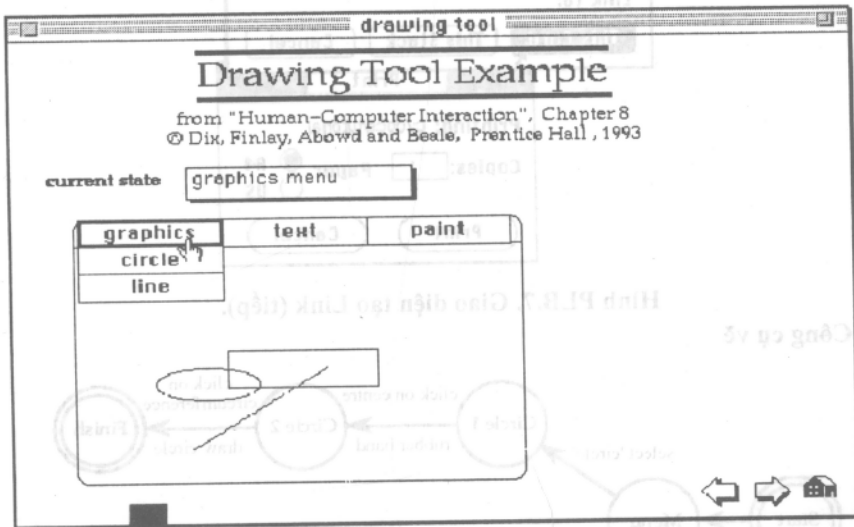
Hình PLB.7. Giao diện tạo Link (tiếp).

Ví dụ - Công cụ vẽ



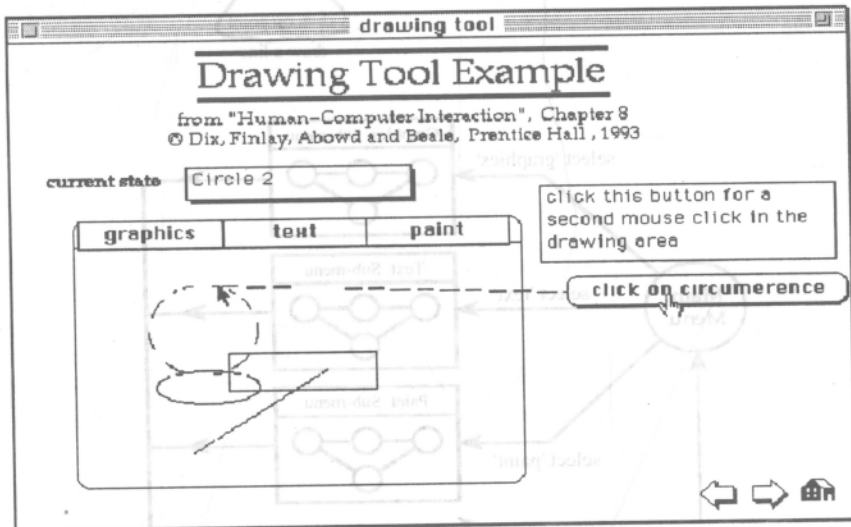
(from Human-Computer Interaction, Dix, Finlay, Abowd and Beale, 1993)

Tạo màn hình cho mỗi trạng thái



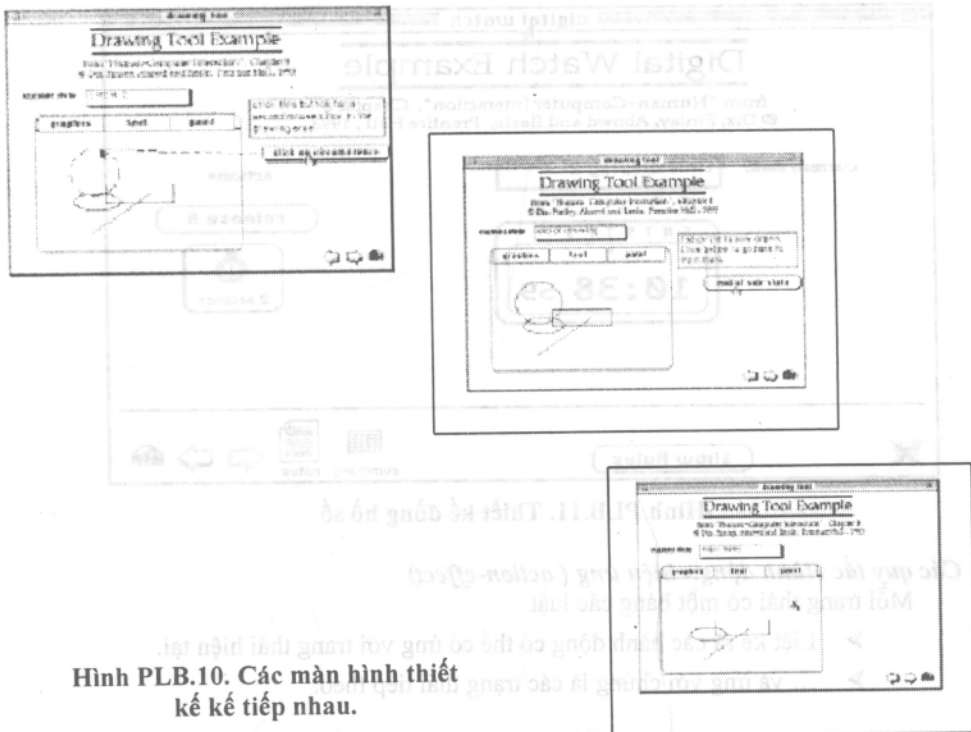
Hình PLB.8. Giao diện thiết kế menu đồ họa.

và thêm các nút vào.



Hình PLB.9. Giao diện thiết kế menu đồ họa (tiếp).

Các màn hình được thiết kế liên tiếp (hình PLB.10)



Hình PLB.10. Các màn hình thiết kế kế tiếp nhau.

3.11. Lấy được nhiều hơn từ dialogue

Có thể kiểm tra trong dialogue

- Tính đầy đủ.
- Tính khả thi.
- Tính đảo ngược.

Nhưng cần sự miêu tả rõ ràng

3.12. Sử dụng để khởi động

- Chỉ rõ dialogue.
- Thực hiện các chi tiết.
- Phân tích các chi tiết.

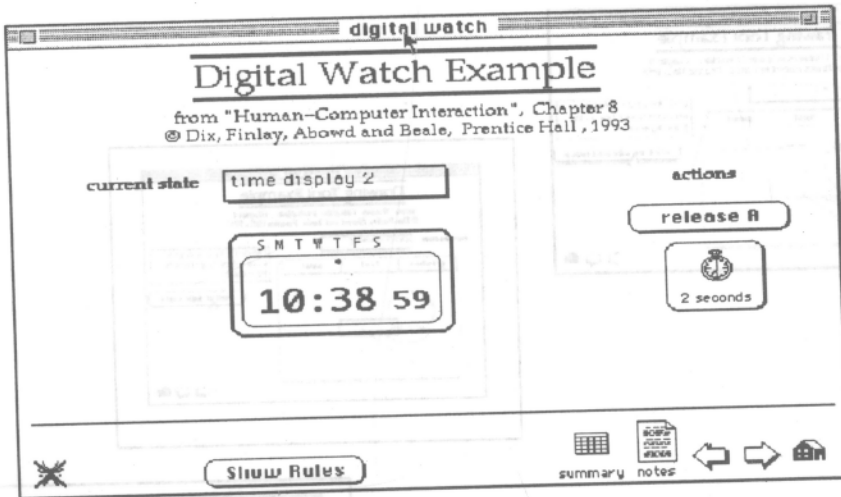
4. Mô phỏng Dialogue

Ví dụ về việc tạo mẫu thử (prototype)

4.1. Tạo dialogue

- Vẽ màn hình bằng tay.
- Tất cả các màn hình (screen) có chung màu nền.
- Các nút thêm vào bằng công cụ.

Phụ lục B: Công cụ tạo lập mẫu thử

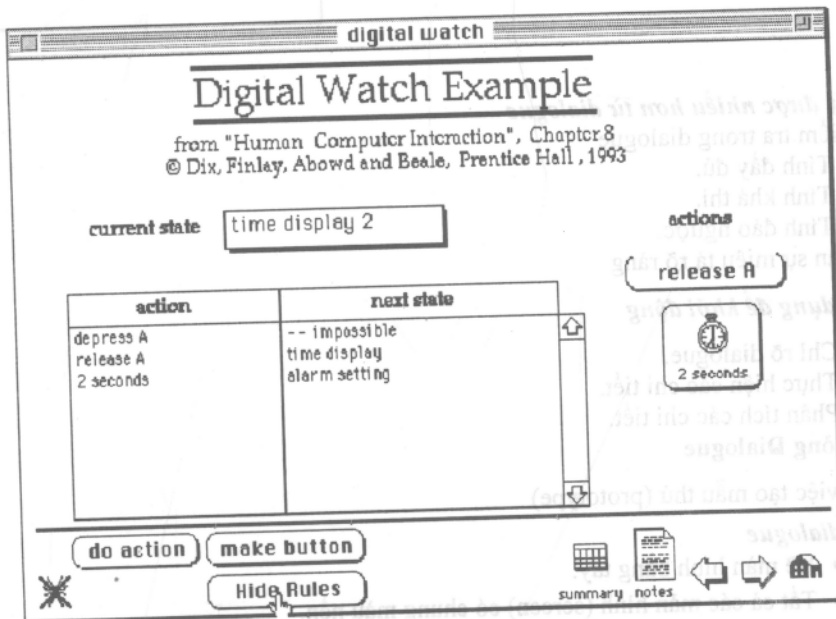


Hình PLB.11. Thiết kế đồng hồ số

4.2. Các quy tắc Hành động - hiệu ứng (action-effect)

Mỗi trạng thái có một bảng các luật

- Liệt kê ra các hành động có thể có ứng với trạng thái hiện tại.
- ... và ứng với chúng là các trạng thái tiếp theo.



Hình PLB.12. Thiết kế đồng hồ số (tiếp).

Phụ lục B: Công cụ tạo lập mẫu thử

4.3. Thực thi các luật

- Được thực hiện khi các nút được nhấn.

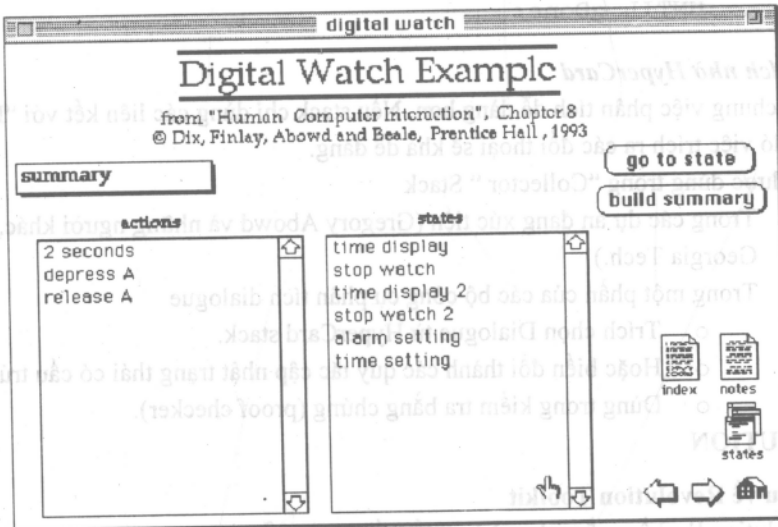


- Thực tế chúng không được kết nối trực tiếp.
- Thay vì bảng ta có thể sử dụng.
 - Tra cứu hành động trong bảng.
 - Trạng thái tương ứng tiếp theo được tìm thấy.
 - Nếu "impossible" hiện thông báo lỗi.
 - Nếu "do nothing" chờ.
 - Nếu không chuyển trạng thái kế tiếp.

4.4. Phân tích các luật

Các tổng kết sẽ liệt kê tất cả:

- Các trạng thái.
- Các hành động trong từng trạng thái.
- Nó có thể kiểm tra.
 - Tất cả các trạng thái tiếp theo (next state) tồn tại.
 - Có thể đạt đến một trạng thái nào đó không.
 - Tất cả các hành động phải có quy tắc trong mỗi trạng thái.



Hình PLB.13. Thiết kế đồng hồ số (tiếp).

4.5. Trích ra các Dialogue

Việc phân tích một chương trình sẵn có có một số đặc điểm sau

- Không cần dùng các ký hiệu đặc biệt (notation).
- Tuy nhiên việc phân tích cực kì khó khăn, do hội thoại và ngữ nghĩa trộn lẫn với nhau.

Ví dụ với một đoạn code như sau:

```
REPEAT
  gotEvent:= WaitNextEvent( everyEvent, myEvent,
15, NIL )
  If NOT DoHandleDialogueEvent(myEvent) THEN
  BEGIN
    CASE myEvent.what OF
      mouseDown: DoMouseDown(myEvent);
      updateEvt:
        DoUpdate(WindowPtr(myEvent.mess
age));
      keyDown, autoKey: DoKeyDown(myEvent);
        etc
      nullEvent: DoIdle(myEvent);
      OTHERWISE: ;
    END; {CASE}
  END
  ELSE
    DoIdle(myEvent);
  UNTIL gDone;
```

4.6. Phân tích nhờ HyperCard

Nói chung việc phân tích dễ dàng hơn. Nếu stack chỉ dùng các liên kết với “Link to...”, khi đó việc trích ra các đối thoại sẽ khá dễ dàng.

Nó được dùng trong “Collector “ Stack

- Trong các dự án đang xúc tiến (Gregory Abowd và những người khác, Georgia Tech.)
- Trong một phần của các bộ công cụ phân tích dialogue
 - Trích chọn Dialogue từ HyperCard stack.
 - Hoặc biến đổi thành các quy tắc cập nhật trạng thái có cấu trúc.
 - Dùng trong kiểm tra bằng chứng (proof checker).

B. REVOLUTION

1. Giới thiệu về Revolution Toolkit

Revolution là phần mềm cho phép thiết kế nhanh mẫu thử giao diện, cũng giống như HyperCard, Revolution cũng cung cấp các công cụ, các palette để thiết kế các Card. Mỗi

Phụ lục B: Công cụ tạo lập mẫu thử

các có thể có các textbox, label, combo box, hay hình ảnh... và thuộc trên một stack mỗi stack có nhiều Card, các Stack tương tác với nhau qua HyperTalk.

Khi Hypercard xuất hiện vào năm 1987, phần lớn mọi người đều không biết phải làm gì với nó. Bill Atkinson, tác giả của Hypercard gọi đó là một công cụ chế tác. Hypercard đã không ngừng phát triển cho thế giới Apple, song cho đến năm 2004 thì nó đã ngừng phát hành. Dream Revolution được phát triển trên ý tưởng của Hypercard và là công cụ được khá kỳ vọng. Revolution hoạt động trên một môi trường khá rộng: Window, Unix, Macs OS.

Revolution có thể thực hiện một số chức năng chính sau:

- **Lập trình:** Revolution gồm một ngôn ngữ lập trình mạnh với hầu hết các khả năng của một ngôn ngữ lập trình như Pascal hay C. Nó cũng có tính chất của lập trình hướng đối tượng.

- **Cơ sở dữ liệu:** Revolution có khả năng thực hiện các lưu trữ, tổ chức, sắp xếp và in ấn với cơ sở dữ liệu nhưng không có ý định thay thế cơ sở dữ liệu. Revolution có thể liên kết với các chương trình cơ sở dữ liệu phức tạp.

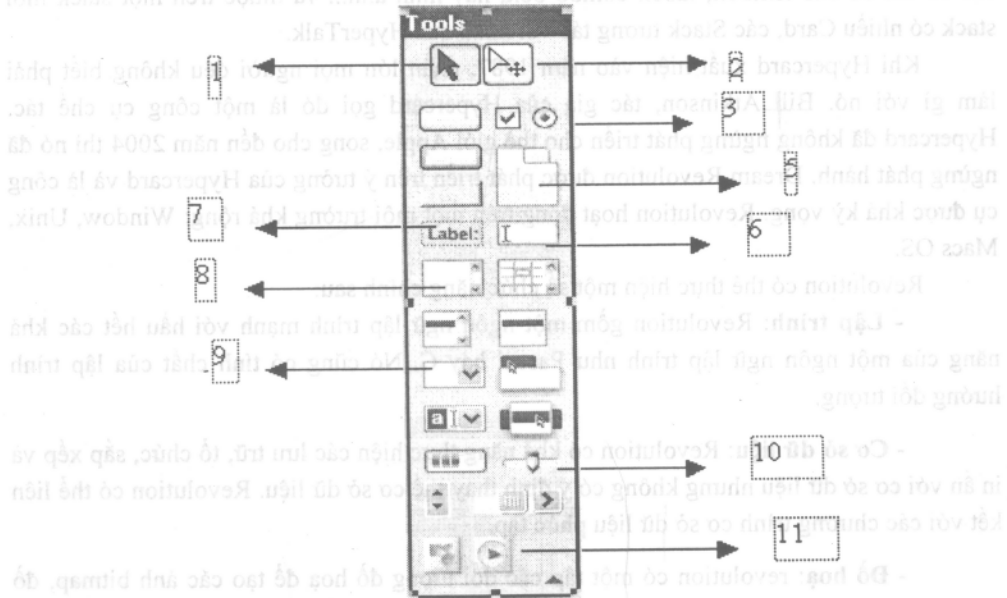
- **Đồ họa:** revolution có một tập các đối tượng đồ họa để tạo các ảnh bitmap, đồ họa như Photoshop. Cảnh trí trong revolution là một ngăn xếp các thẻ chỉ số. Mỗi thẻ có thông tin trên đó. Stack là chồng các thẻ, thường là các thẻ tương tự hay có liên quan. Mỗi thẻ có thể chứa các đối tượng: đồ họa, văn bản, phím lệnh, đa phương tiện và menu (hình PLB14).

Đối tượng cơ bản khi xây dựng ứng dụng của revolution là Stack. Mọi cửa sổ nhìn thấy trong môi trường đều là Stack. Không có đối tượng nào tồn tại độc lập với Stack. Do vậy, bước đầu tiên để tạo một ứng dụng là tạo một stack (sử dụng Menu File- New Maintstack).

Đối tượng thứ hai là Card. Mọi Stack hiển thị thông tin bằng một hay nhiều card.. Ngầm định, mỗi stack chứa ít nhất một card và do vậy khi một stack được tạo thì một card cũng được tạo ra. Để thêm card vào stack, ta chọn New Card trong menu Object. Tuy là một stack có nhiều card, song tại một thời điểm chỉ có một card được hiện.

Nhóm là kiểu đối tượng thứ ba của revolution. Nhóm có điểm giống card là nó chứa tất cả các đối tượng và tạo ra điều khiển nhóm. Các nhóm và điều khiển nhóm có thể đặt trên các card.

Tất cả các đối tượng khác được chứa trên các card theo các cách khác nhau. Các đối tượng chủ yếu là các phím lệnh (Button), trường (field), ảnh (image), đồ họa (graphic) và các player.



Hình PLB.14. Panel điều khiển của Revolution Toolkit.

- 1: chế độ run;
- 2: chế độ design;
- 3: các điều khiển checkbox và radiobox;
- 5: các nút và các tab;
- 6: hộp textbox;
- 7: label;
- 8: list box;
- 9: điều khiển combo Box;
- 10: các thanh tiến trình, thanh trượt;
- 11: các điều khiển Multimedia.

2. Làm việc với Revolution

Thường để thiết kế chương trình với revolution, người ta có thể sử dụng một trong các câu trúc sau:

- Tuyển tính
- Nhảy tuyển tính (Jump linear)

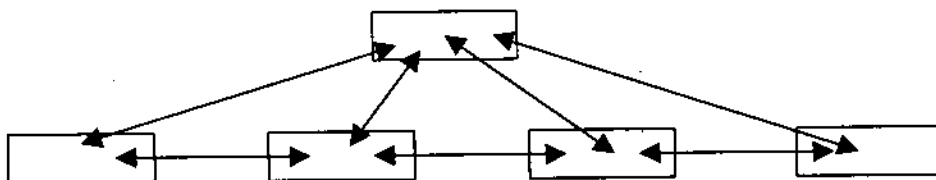
Phụ lục B: Công cụ tạo lập mẫu thử

- Nhánh chặt chẽ (Strict branching)
- Nhánh mở (Open branching)
- Chỉ dẫn hay siêu liên kết

2.1. Cấu trúc tuyến tính

Cấu trúc tuyến tính là cấu trúc đơn giản nhất (hình PLB -15). Trong revolution nó thường bao gồm:

- Một mainstack.
- Một nhóm chứa tất cả các trường.
- Một card menu không có nhóm.
- Có đủ số card cần thiết với nhóm trên tất cả các card còn lại.

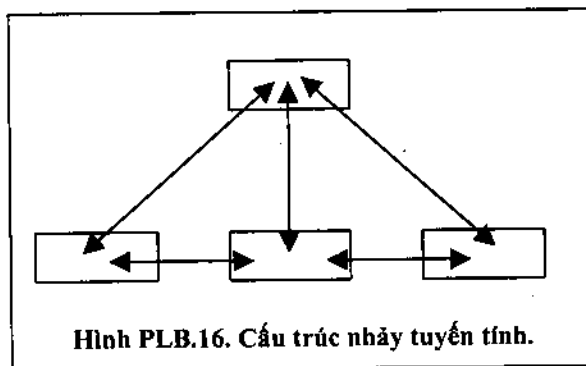


Hình PLB.15. Cấu trúc tuyến tính.

2.2. Cấu trúc nhảy tuyến tính

Cấu trúc này bao gồm:

- mainstack;
- 3 substacks;
- một nhóm trong mỗi substack chứa tất cả các trường và phím lệnh;
- Có đủ số card cần thiết trên mỗi stack với nhóm trên mỗi stack.

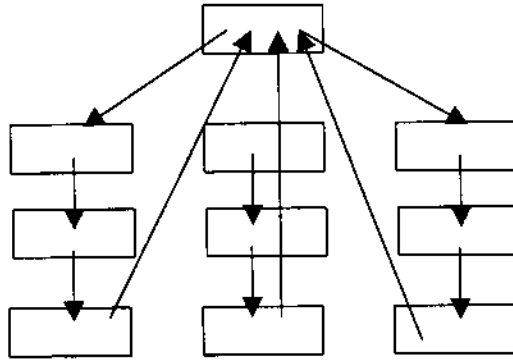


Hình PLB.16. Cấu trúc nhảy tuyến tính.

2.3. Cấu trúc nhánh chặt

Cấu trúc này bao gồm:

- mainstack;
- 3 substacks;
- một nhóm trong mỗi substack chứa tất cả các trường và phim lệnh;
- có đủ số card cần thiết trên mỗi substack với nhóm trên mỗi stack.

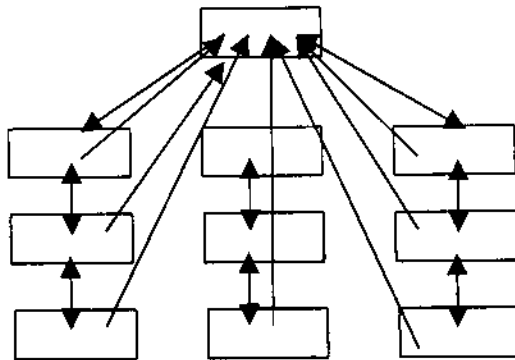


Hình PLB.17. Cấu trúc nhánh chặt.

2.4. Cấu trúc nhánh mở

Cấu trúc này bao gồm:

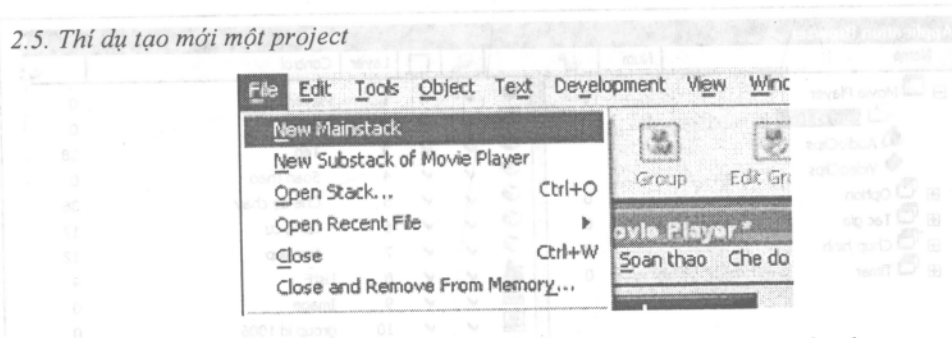
- mainstack;
- 3 substacks;
- một nhóm trong mỗi substack chứa tất cả các trường và phim lệnh;
- có đủ số card cần thiết trên mỗi substack với nhóm trên mỗi stack.



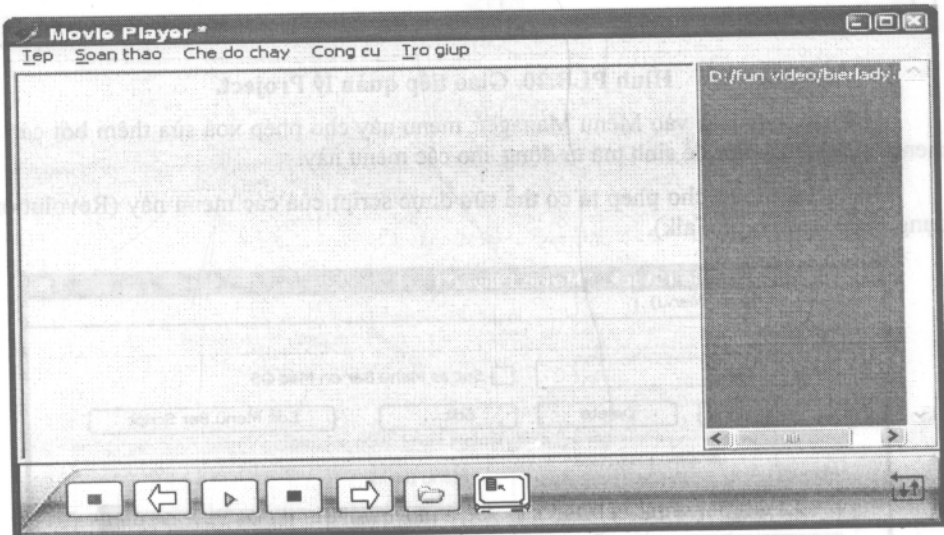
Hình PLB.18. Cấu trúc nhánh mở.

Phụ lục B: Công cụ tạo lập mẫu thử

2.5. Thí dụ tạo mới một project

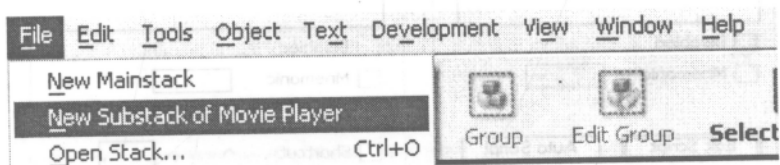


Vào cửa sổ file tạo chọn New Mainstack: Ta bắt đầu làm việc thiết kế với Mainstack này với giao diện rất trực quan và gần gũi, tạo được một mainstack như sau (hình PLB.19):



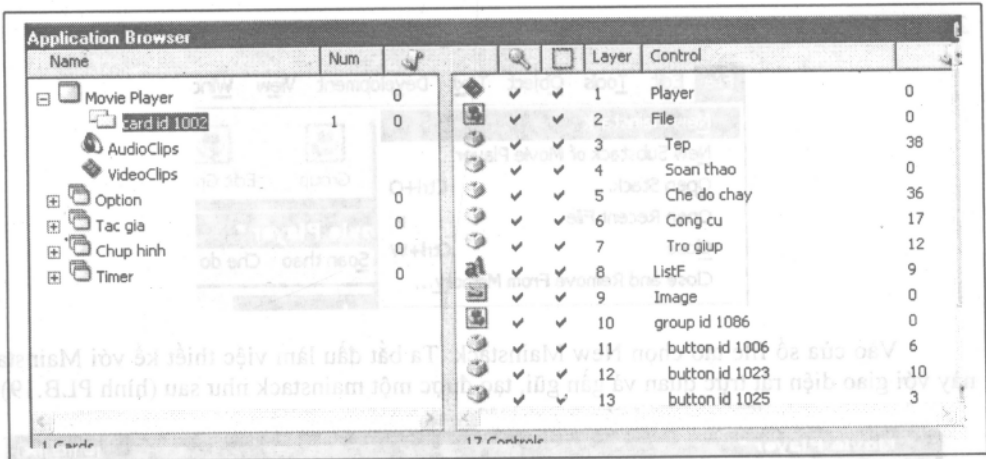
Hình PLB.19. Tạo Mainstack cho một ứng dụng Movie Player.

Để tạo được các Submainstack ta vào File\New SubStack of..., ta sẽ được các stack con của stackmain.



Để quản lý được toàn bộ Projects ta sử dụng cửa sổ Application Browser. Ta dễ dàng có thể thấy rõ được cấu trúc thư mục của Project:

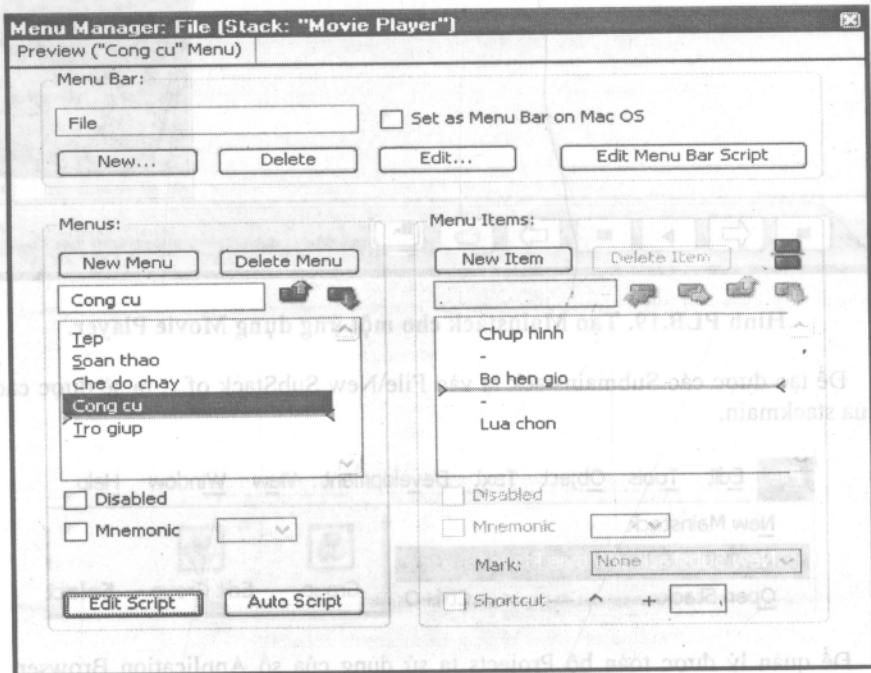
Phụ lục B: Công cụ tạo lập mẫu thử



Hình PLB.20. Giao tiếp quản lý Project.

Để tạo menu, ta vào Menu Manager: menu này cho phép xoá sửa thêm bớt các mục menu, nút AutoScript để sinh mã tự động cho các menu này.

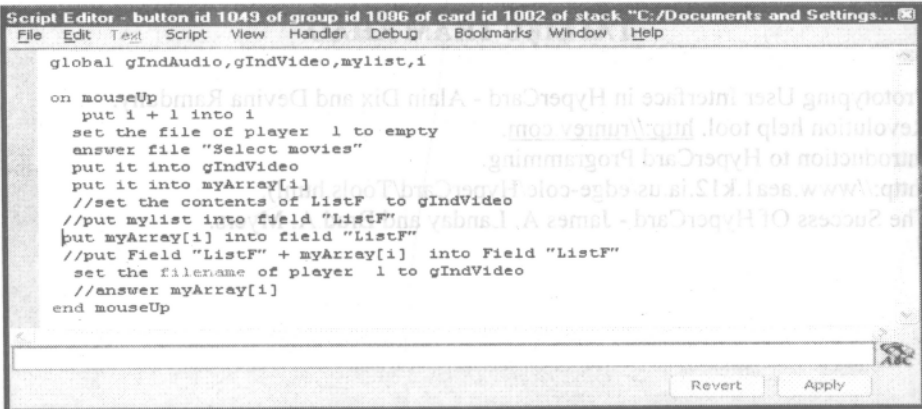
Nút EditScript cho phép ta có thể sửa được script của các menu này (Revolution sử dụng ngôn ngữ HyperTalk).



Hình PLB.21. Giao tiếp của menu manager.

Phụ lục B: Công cụ tạo lập mẫu thử

Với mỗi một đối tượng ta đều có một cửa sổ soạn thảo kịch bản riêng:



```
Script Editor - button id 1049 of group id 1086 of card id 1002 of stack "C:/Documents and Settings...
File Edit Text Script View Handler Debug Bookmarks Window Help

global gIndAudio,gIndVideo,mylist,1

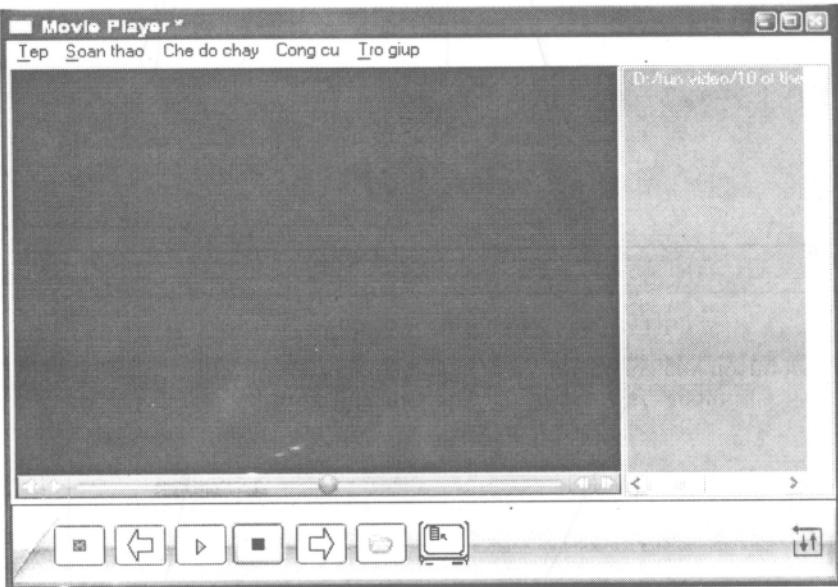
on mouseUp
  put 1 + 1 into i
  set the file of player 1 to empty
  answer file "Select movies"
  put it into gIndVideo
  put it into myArray[1]
  //set the contents of ListF to gIndVideo
  //put mylist into field "ListF"
  put myArray[1] into field "ListF"
  //put Field "ListF" + myArray[1] into Field "ListF"
  set the filename of player 1 to gIndVideo
  //answer myArray[1]
end mouseUp

Revert Apply
```

Hình PLB.22. Cửa sổ soạn thảo mã lệnh Script.

Ứng dụng Movie Player trên không những chỉ mô phỏng mà còn hoạt động được, nó có thể play được một số định dạng media như mpeg, avi, mp3, wav, audio file...

Ngoài ra nó còn có một số chức năng như SoundRecorder, save file đang chạy sang một nơi khác, chụp hình và save dưới dạng một file ảnh, ngoài ra nó còn mô phỏng giống như là một chương trình xem phim nghe nhạc thực thụ, hơn nữa nó có thể chạy trên đa nền hệ điều hành như Windows, OS X, Unix tùy vào yêu cầu sử dụng của người thiết kế (hình PLB.23).



Hình PLB.23. Ứng dụng Movie Player xây dựng trên revolution.

TÀI LIỆU THAM KHẢO

- [1] Prototyping User Interface in HyperCard - Alain Dix and Devina Ramduny.
- [2] Revolution help tool. <http://runrev.com>.
- [3] Introduction to HyperCard Programming.
(<http://www.aea1.k12.ja.us/edge-cole/HyperCard/Tools.html>).
- [4] The Success Of HyperCard - James A. Landay and Brad A. Myers.

Phụ lục C

CÁC CHUẨN ISO

I. TỔNG QUAN VỀ ISO VÀ CÁC CHUẨN ISO 9241, ISO 11581

1. Đôi nét về tổ chức ISO

ISO là viết tắt của International Organisation for Standardisation và là một hệ thống các Viện, Trường chuẩn quốc gia từ 147 nước trên thế giới. Do công nghệ là thương mại quốc tế nên các nhà sản xuất rất chú ý vào các chuẩn quốc tế. Rất nhiều các quốc gia nhất là châu Âu cũng đã chấp nhận các chuẩn ISO như là chuẩn quốc gia của mình. Các chuẩn được áp dụng trong nhiều ngành, nhiều lĩnh vực khác nhau. Ở đây, ta chỉ trình bày một số chuẩn liên quan đến việc tương tác người - máy như ISO 9241 liên quan đến tính tiện dùng (usability), ISO 11581 liên quan đến thiết kế các đối tượng tương tác.

2. Giới thiệu về chuẩn ISO 9241

ISO 9241 là một trong rất nhiều chuẩn về tính dùng được và lĩnh vực công thái học trong lao động học (ergonomics - the study of the relationship between people and their working environment - nghiên cứu về mối quan hệ giữa con người và môi trường làm việc của họ).

Chuẩn ISO 9241 được trích dẫn nhiều hơn là được đọc vì nó chứa nhiều thông tin về mọi khía cạnh của tính tiện dùng, bao gồm phần cứng (hardware), phần mềm (software) và các quy trình có tính tiện dùng. Có thể sử dụng chuẩn này để thiết kế một trạm làm việc, lượng giá một màn hình, thiết lập các độ đo tính tiện dùng, lượng giá một giao diện đồ họa, áp dụng một bàn phím mới, thẩm định một thiết bị tương tác mới như joystick, kiểm tra xem môi trường làm việc đã được chưa, đo độ phản xạ và màu sắc của một màn hình hiển thị,... Nó chứa các bản liệt kê các mục cần kiểm tra giúp cho việc cấu trúc một lượng giá về tính tiện dùng. Ví dụ làm thế nào để vận hành và đo tính tiện dùng và các mục có phạm vi rộng khác.

3. Giới thiệu về chuẩn ISO/IEC 11581

ISO 11581 gồm có 6 phần:

- Phần 1: Icons - General. Trình bày tổng quan và về biểu tượng.
- Phần 2: Object Icons. Liên quan đến thiết kế các biểu tượng.
- Phần 3: Pointer Icons. Liên quan đến thiết kế các dạng con trỏ.
- Phần 4: Control Icons. Liên quan đến thiết kế các biểu tượng điều khiển.
- Phần 5: Tool Icons. Liên quan đến thiết kế các biểu tượng các thanh công cụ.
- Phần 6: Action icons. Liên quan đến thiết kế các biểu tượng hành động.

Các icon (biểu tượng) được sử dụng trên màn hình để thực hiện sự tương tác giữa các ứng dụng và người dùng. Các biểu tượng đồ họa có thể cung cấp sự độc lập về ngôn ngữ trong khi trao đổi thông tin với người sử dụng. Chúng là một phần của giao diện đồ họa có thể tận dụng khả năng học, hiểu và ghi nhớ các phần tử chức năng của hệ thống và trợ giúp trong khi thao tác với những phần tử đó.

Thông thường, một giao diện đồ họa người dùng - GUI dựa trên môi trường để cung cấp một biểu diễn ẩn dụ cho các nhiệm vụ của người dùng. Một ẩn dụ cung cấp phép suy luận tới các khái niệm đã quen thuộc với người dùng, từ đó người dùng có thể suy diễn ra các trường hợp sử dụng và các hành vi của hệ thống. Các biểu tượng có thể biểu diễn các ẩn dụ một cách trực tiếp, hay cũng có thể biểu diễn một đối tượng vật lý một cách trực tiếp.

Các biểu tượng này được phân biệt với các biểu tượng khác nhờ các chức năng hệ thống mà chúng thực hiện. Các biểu tượng biểu diễn các đối tượng, các con trỏ, và các công cụ tạo nên một miền ứng dụng và những thứ mà người dùng thao tác trong khi thực hiện công việc của họ. Họ cũng có thể biểu diễn các chỉ thị trạng thái dùng bởi hệ thống máy tính để đưa ra các thông tin tới người dùng và để dàn xếp các tương tác của người dùng với ứng dụng phần mềm.

II. CHUẨN ISO 9241

ISO 9241 gồm có 17 phần. Dưới đây sẽ trình bày tóm tắt về mỗi phần trong ISO 9241. Mỗi phần chứa các thông tin sau:

- *Citation*: cung cấp các tham chiếu đầy đủ mà ta nên sử dụng khi trích dẫn tới chuẩn trong một tài liệu.
- *Status*: mô tả vị trí hiện thời của chuẩn. Các chuẩn quốc tế được hình thành qua một vài giai đoạn từ khi là bản thảo cho tới khi được công nhận ở phạm vi quốc tế. Lịch sử phát triển của một chuẩn ISO gồm có 6 giai đoạn:
 - AWI: Approved Work Item
 - WD: Working Draft
 - CD: Committee Draft
 - DIS: Draft International Standard
 - FDIS: Final Draft International Standard
 - ISO: International Standard
- *Lifecycle phase*: mô hình vòng đời có 4 pha là **Analyse the opportunity, Build the context of use, Create the user experience, Track real-world usage and continuously improve the system**:
 - Analyse the opportunity - phân tích cơ hội: pha này cung cấp ngữ cảnh nghiệp vụ cho sản phẩm hoặc dịch vụ. Trong pha này, các chuyên gia về tính tiện dùng xác định xem tại sao hệ thống lại đang được phát triển, xác định các cổ đông (stakeholder) và phân đoạn thị trường cho hệ thống. Họ tìm hiểu các nền (platform) và các kênh điện tử (e-channel) với trọng tâm là thái độ và hành vi của audience.

Phụ lục C: Các chuẩn trong thiết kế HCI

- Build the context of use - xây dựng ngữ cảnh sử dụng: Pha này cho kết quả là bản mô tả chi tiết về khách hàng, môi trường mà khách hàng truy cập hệ thống và một bản mô tả các hoạt động thực (các cảnh). Các chuyên gia về tính tiện dùng sử dụng các bản mô tả đó để lấy ra được các khía cạnh tâm lý học vốn có từ kinh nghiệm khách hàng.
 - Create the user experience - tạo ra kinh nghiệm người sử dụng: đây là tiến trình lặp và ngắn. Các chuyên gia về tính tiện dùng bắt đầu thoả thuận về các chỉ số hiệu suất chính của hệ thống: các độ đo định lượng, dựa trên các khách hàng chính và các yêu cầu nghiệp vụ (đội dự án sử dụng nó để xác định xem liệu hệ thống đã sẵn sàng phát hành được chưa). Tiếp theo, họ phát triển kiến trúc giao diện người sử dụng (thiết kế mức khái niệm, mức cao) và thực hiện thiết kế chi tiết, bắt đầu với phác thảo trên giấy và sau đó chuyển sang các bản minh hoạ điện tử (electronic slide shows) và các prototype tương tác. Tiếp đó, kiểm tra tính sử dụng bởi các chuyên gia về lĩnh vực và các khách hàng tiêu biểu.
 - Track real-world usage and continuously improve the system - theo dõi việc sử dụng trong thực tế và tiếp tục cải tiến hệ thống: các chuyên gia về tính tiện dùng theo dõi hành vi của khách hàng khi họ cập nhật các kỹ năng, các nhiệm vụ và môi trường của họ. Họ cũng kiểm tra một số độ đo nghiệp vụ chính để đảm bảo rằng các khách hàng có thu được kết quả từ sự đầu tư của họ.
- Type of guidance - kiểu hướng dẫn: một số phần trong ISO 9241 chứa các đặc tả chi tiết. Ví dụ, ISO 9241-3 chứa các đặc tả sau: " Với kiểu phông không chân, khoảng cách giữa các ký tự nên tối thiểu là một lần nhân hay 1 pixel. ". Các phần khác chứa các hướng dẫn chung. Ví dụ, ISO 9241-10 chứa khuyến nghị sau: "Nếu tồn tại các ngầm định với một nhiệm vụ đã cho thì người dùng phải biết". Phần này đơn giản chỉ xác định kiểu hướng dẫn của chuẩn.
 - Application area - vùng ứng dụng: một số phần trong ISO 9241 áp dụng cho phần cứng, một số khác thì cho phần mềm và số khác nữa thì áp dụng cho các quy trình có tính tiện dụng. Các hardware guidelines có ích trong việc đánh giá tính lao động học của các máy trạm. Các software guidelines có ích cho hướng dẫn chi tiết về cách thức mà các giao diện người dùng nên dùng theo kiểu "look and feel". Quá trình hướng dẫn có ích cho việc xác định chất lượng tổng thể và các yêu cầu về tính dùng được. Phần này xác định vùng ứng dụng cho chuẩn.
 - Scope: Cung cấp mô tả chi tiết cho các nội dung của chuẩn.
 - Audience: ISO 9241 nhằm vào một loạt các nhóm chuyên gia như: Các kỹ sư, các nhà chuyên nghiệp về tính tiện dụng, các nhà thiết kế và phát triển giao diện người dùng, người dùng cuối, và những người chịu trách nhiệm thu mua các phần cứng và phần mềm. Tuy nhiên, một vài phần yêu cầu một chút thông thạo về kỹ thuật. Các phần khác là dành cho mọi đối tượng. Trong phần này sẽ đưa ra một vài ý tưởng về các kỹ năng công nghệ mà người đọc cần có để hiểu được chuẩn.

Phụ lục C: Các chuẩn trong thiết kế HCI

- See also: chứa các tham khảo chéo tới các phần khác của chuẩn ISO 9241 cũng như các chuẩn khác.

1. General Introduction

CITATION: ISO 9241-1:1997 Các yêu cầu lao động học cho công việc văn phòng với visual display terminals (VDTs).

STATUS

AWI WD CD DIS FDIS ISO

LIFECYCLE PHASE

Analyse the opportunity Create the user experience

Build the context of use Track usage and improve

TYPE OF GUIDANCE

Specifications Principles and general recommendations

APPLICATION AREA

General guidance Equipment Environme

Software User Testing Process

SCOPE: có vai trò như là phần dẫn nhập cho tất cả các phần trong chuẩn ISO 9241. Nó mô tả mục đích của chuẩn 9241 và thảo luận về triết lý hiệu suất người dùng. Triết lý này là nền tảng của chuẩn ISO 9241. Việc tạo ra các chuẩn là một tiến trình khá chậm, một phần là do yêu cầu về sự thống nhất cao và một phần là do phải mất thời gian để tính ổn định có thể hiện rõ trong bất kỳ một công nghệ mới nào. Điều này nghĩa là các chuẩn về tính lao động học có thể chưa có sẵn để hỗ trợ cho việc mua các thiết bị giao diện mới hơn. Trong những trường hợp như vậy, việc yêu cầu các nhà sản xuất chứng minh cho tính tiện dùng của các sản phẩm của họ sẽ cung cấp lộ trình hiệu quả nhất, đảm bảo cho chất lượng lao động học tốt. Vì thế một số phần trong chuẩn ISO 9241 chưa có các kiểm thử hiệu suất người dùng, để giúp cho các nhà sản xuất thực hiện điều đó. Chuẩn cũng bao gồm một tiểu sử (bibliography) và một phụ lục thông tin mô tả cách thức áp dụng các chuẩn cho phần mềm của ISO 9241 (phần 10-17).

AUDIENCE: tất cả mọi đối tượng quan tâm đối ISO 9241.

2. Guidance on task requirements

CITATION: ISO 9241-2:1992 Các yêu cầu lao động học cho công việc văn phòng với màn hình hiển thị.

STATUS:

AWI WD CD DIS FDIS ISO

Phụ lục C: Các chuẩn trong thiết kế HCI

LIFECYCLE PHASE

- | | |
|--|---|
| <input type="checkbox"/> Analyse the opportunity | <input type="checkbox"/> Create the user experience |
| <input checked="" type="checkbox"/> Build the context of use | <input type="checkbox"/> Track usage and improve |

TYPE OF GUIDANCE

- | | |
|---|--|
| <input type="checkbox"/> Specifications | <input checked="" type="checkbox"/> Principles and general recommendations |
|---|--|

APPLICATION AREA

- | | | |
|--|---------------------------------------|--------------------------------------|
| <input checked="" type="checkbox"/> General guidance | <input type="checkbox"/> Equipment | <input type="checkbox"/> Environment |
| <input type="checkbox"/> Software | <input type="checkbox"/> User Testing | <input type="checkbox"/> Process |

SCOPE: phần này cung cấp các hướng dẫn cho việc thiết kế các tác vụ và các job liên quan đến công việc phải tương tác với màn hình. Nó cung cấp hướng dẫn về cách xác định các yêu cầu tác vụ và cách xác định chúng trong một tổ chức. Nó cũng mô tả cách thức mà các yêu cầu tác vụ có thể được kết hợp vào thiết kế hệ thống và tiến trình cài đặt.

AUDIENCE: các nhà quản lý và các nhà thiết kế chịu trách nhiệm tổ chức các công việc thực tế.

3. Visual display requirements

CITATION: ISO 9241-3: 1992 Các yêu cầu lao động học cho công việc văn phòng với màn hình.

STATUS:

- | | | | | | |
|------------------------------|-----------------------------|-----------------------------|------------------------------|-------------------------------|---|
| <input type="checkbox"/> AWI | <input type="checkbox"/> WD | <input type="checkbox"/> CD | <input type="checkbox"/> DIS | <input type="checkbox"/> FDIS | <input checked="" type="checkbox"/> ISO |
|------------------------------|-----------------------------|-----------------------------|------------------------------|-------------------------------|---|

LIFECYCLE PHASE

- | | |
|--|---|
| <input type="checkbox"/> Analyse the opportunity | <input type="checkbox"/> Create the user experience |
| <input checked="" type="checkbox"/> Build the context of use | <input type="checkbox"/> Track usage and improve |

TYPE OF GUIDANCE

- | | |
|--|---|
| <input checked="" type="checkbox"/> Specifications | <input type="checkbox"/> Principles and general recommendations |
|--|---|

APPLICATION AREA

- | | | |
|---|--|--------------------------------------|
| <input type="checkbox"/> General guidance | <input checked="" type="checkbox"/> Equipment | <input type="checkbox"/> Environment |
| <input type="checkbox"/> Software | <input checked="" type="checkbox"/> User Testing | <input type="checkbox"/> Process |

SCOPE: phần này thiết lập các yêu cầu chất lượng hình ảnh cho thiết kế và sự đánh giá cho

Phụ lục C: Các chuẩn trong thiết kế HCI

TYPE OF GUIDANCE

Specifications Principles and general recommendations

APPLICATION AREA

General guidance Equipment Environment
 Software User Testing Process

SCOPE: phần này cung cấp các hướng dẫn về các đặc trưng cơ bản của môi trường làm việc để hỗ trợ cho nghiệp vụ sử dụng đến màn hình hiển thị. Các đặc tính của môi trường làm việc gồm sáu tính chất sau: ánh sáng tự nhiên và nhân tạo; âm thanh và tiếng ồn; các dao động cơ học; trường điện từ và tĩnh điện; độ ẩm môi trường; không gian tổ chức và sơ đồ phân bố. Chuẩn bao gồm 4 phụ lục sau: ánh sáng, các phương pháp đo và định lượng âm thanh; measurement, evaluation and assessment of whole body vibrations; và độ ẩm môi trường.

AUDIENCE: những người quan tâm đến việc thiết kế và thực hiện bố trí văn phòng.

7. Requirements for display with reflections

CITATION: ISO 9241-7:1998 Các yêu cầu lao động học cho công việc văn phòng tương tác với màn hình.

STATUS

AWI WD CD DIS FDIS ISO

LIFECYCLE PHASE

Analyse the opportunity Create the user experience
 Build the context of use Track usage and improve

TYPE OF GUIDANCE

Specifications Principles and general

APPLICATION AREA

General guidance Equipment Environment
 Software User Testing Process

SCOPE: phần này có nhiệm vụ trợ giúp các nhà thiết kế để đo sự phản xạ từ màn hình mà có ảnh hưởng đến người dùng và hiệu suất công việc khi sử dụng các thiết bị hiển thị. Nó xác định các yêu cầu và các phương pháp để đo chất lượng hình ảnh trong ánh sáng môi trường mà có thể gây ra sự phản chiếu và phản xạ khuếch tán từ màn hình. Các yêu cầu bao gồm: tỷ lệ giữa độ chói và độ phản xạ; specular reflection luminance ratio; phương pháp

Phụ lục C: Các chuẩn trong thiết kế HCI

9. Requirements for non-keyboard input devices

CITATION:ISO 9241-9:2000 Các yêu cầu lao động học cho công việc văn phòng tương tác với màn hình.

STATUS

AWI WD CD DIS FDIS ISO

LIFECYCLE PHASE

Analyse the opportunity Create the user experience
 Build the context of use Track usage and improve

TYPE OF GUIDANCE

Specifications Principles and general

APPLICATION AREA

General guidance Equipment Environment
 Software User Testing Process

SCOPE: phần này áp dụng cho vài kiểu thiết bị ngoại vi như: chuột, pucks, joysticks, trackballs, tablets và overlays; touch-sensitive screens, style and light pens. Chuẩn xác định chất lượng của thiết bị nhập theo tiêu chuẩn hiệu suất. Chuẩn cũng bao gồm một tập các yêu cầu thiết kế mà đầu tiên đưa ra các yêu cầu chung và các khuyến nghị, sau đó tập trung vào xác định các yêu cầu của thiết bị nhập và các khuyến nghị (như chuột, pucks, và joysticks). Sự tuân theo chuẩn chỉ có thể thực hiện được bằng cách tiến hành các kiểm thử về tính dùng được. Chuẩn có 4 phụ lục: lựa chọn các thiết bị nhập, phân tích và kiểm tra tính tiện dùng; kiểm thử hiệu năng và hiệu quả; thẩm định tính tiện nghi; và các phương pháp định giá phụ.

AUDIENCE: các nhà sản xuất thiết bị nhập khác bàn phím và những người cần kiểm tra các thiết bị này.

10. Dialogue Principles

CITATION:ISO 9241-10:1996 Các yêu cầu lao động học cho công việc văn phòng tương tác với màn hình.

STATUS

AWI WD CD DIS FDIS ISO

LIFECYCLE PHASE

Analyse the opportunity Create the user experience
 Build the context of use Track usage and improve

Phụ lục C: Các chuẩn trong thiết kế HCI

TYPE OF GUIDANCE

Specifications Principles and general recommendations

APPLICATION AREA

General guidance Equipment Environment

Software User Testing Process

SCOPE: phần này của chuẩn ISO 9241 biểu diễn một tập usability heuristics để áp dụng cho sự tương tác giữa con người và các hệ thống thông tin. Chuẩn này gọi sự tương tác này như một "dialogue" và mô tả bảy nguyên tắc đối thoại. Các nguyên tắc chung này mở rộng các kỹ thuật hội thoại rõ ràng. Bảy nguyên tắc đó là: hội thoại nên phù hợp với các nhiệm vụ và kỹ năng của người dùng; tính tự mô tả (nên chỉ rõ cho người dùng nên làm việc gì tiếp theo); người dùng nên có khả năng điều khiển nhịp độ và tuân tự của các tương tác; phù hợp với mong đợi của người dùng; dung lỗi; có khả năng tùy biến cho người dùng) và hỗ trợ cho việc học). Chuẩn này mô tả các ứng dụng và các ví dụ của các nguyên tắc hội thoại.

AUDIENCE: những người thiết kế giao diện và những người đánh giá.

11. Guidance on usability

CITATION: ISO 9241-11:1998 Các yêu cầu lao động học cho công việc văn phòng tương tác với màn hình.

STATUS

AWI WD CD DIS FDIS ISO

LIFECYCLE PHASE

Analyse the opportunity Create the user experience

Build the context of use Track usage and improve

TYPE OF GUIDANCE

Specifications Principles and general recommendations

APPLICATION AREA

General guidance Equipment Environment

Software User Testing Process

SCOPE: phần này giới thiệu khái niệm về tính dùng được nhưng không đưa ra các khuyến nghị cụ thể về các thuộc tính của sản phẩm. Thay vào đó, nó đưa ra các định nghĩa về tính tiện dùng như một sự mở rộng tới các sản phẩm có thể được sử dụng bởi những người dùng

Phụ lục C: Các chuẩn trong thiết kế HCI

cụ thể để thực hiện nhiệm vụ hiệu quả, hiệu suất và thoải mái trong một ngữ cảnh sử dụng cụ thể. Một ưu điểm của cách tiếp cận này là nó giúp cho các đội thiết kế dự kiến về tính dùng được như là một phần của chu kỳ phát triển, bằng cách xác lập và đo tính tiện dùng cho sản phẩm. Chuẩn này mô tả cách thức mà nó được áp dụng để: xác định và đo mục tiêu tính tiện dùng của sản phẩm; xác định và lượng giá tính tiện dùng trong khi thiết kế; xác định và đo một hệ thống công việc đang được dùng. Chuẩn cũng bao gồm 5 phụ lục sau: ví dụ về cách thức xác định ngữ cảnh sử dụng; ví dụ về sự đo tính tiện dùng; ví dụ về một đặc tả yêu cầu về tính tiện dùng; mối quan hệ giữa các chuẩn quốc tế khác nhau và một thư viện AUDIENCE: những người cần thiết lập và đo mục tiêu dùng được của một hệ thống.

12. Presentation of information

CITATION: ISO 9241-12:1998 Các yêu cầu lao động học cho công việc văn phòng tương tác với màn hình.

STATUS

AWI WD CD DIS FDIS ISO

LIFECYCLE PHASE

Analyse the opportunity Create the user experience
 Build the context of use Track usage and improve

TYPE OF GUIDANCE

Specifications Principles and general recommendations

APPLICATION AREA

General guidance Equipment Environment
 Software User Testing Process

SCOPE: phần này chứa các khuyến nghị về cách thức biểu diễn các thông tin trực quan trên màn hình sao cho người dùng có thể dễ dàng thực hiện các nhiệm vụ cảm nhận - perceptual tasks. Các khuyến nghị được dựa trên bảy nguyên tắc sau đây: rõ ràng ; có tính phân biệt ; ngắn gọn; nhất quán; phát hiện được; dễ đọc và dễ hiểu.

Các khuyến nghị được cung cấp trên ba lĩnh vực chính: tổ chức thông tin; các đối tượng đồ họa; các kỹ thuật mã hoá. Không có mô tả về thiết kế icon. Có hai phụ lục.

AUDIENCE: những người thiết kế giao diện và những người đánh giá.

13. User guidance

CITATION: ISO 9241-13:1998 Các yêu cầu lao động học cho công việc văn phòng tương tác với màn hình.

Phụ lục C: Các chuẩn trong thiết kế HCI

STATUS

AWI WD CD DIS FDIS ISO

LIFECYCLE PHASE

Analyse the opportunity Create the user experience
 Build the context of use Track usage and improve

TYPE OF GUIDANCE

Specifications Principles and general recommendations

APPLICATION AREA

General guidance Equipment Environment
 Software User Testing Process

SCOPE: phần này bao gồm các khuyến nghị về hướng dẫn người dùng. Các khuyến nghị đưa ra một vài lời khuyên chung; các lời nhắc; phản hồi từ user; trạng thái thông tin; quản lý lỗi; trợ giúp trực tuyến. Chuẩn này không có bao gồm tài liệu hoặc bài học trực tuyến. Chuẩn chứa hai phụ lục. Đầu tiên là một thủ tục mẫu để thẩm định tính áp dụng được, tôn trọng và bao gồm 6 page checklist. Thứ hai là một thư viện.

AUDIENCE: các nhà thiết kế giao diện, các nhà đánh giá giao diện và các chuyên gia về làm tài liệu.

14. Menu dialogues

CITATION: ISO 9241-14:1997 Các yêu cầu lao động học cho công việc văn phòng tương tác với màn hình.

STATUS

AWI WD CD DIS FDIS ISO

LIFECYCLE PHASE

Analyse the opportunity Create the user experience
 Build the context of use Track usage and improve

TYPE OF GUIDANCE

Specifications Principles and general recommendations

APPLICATION AREA

General guidance Equipment Environment
 Software User Testing Process

Phụ lục C: Các chuẩn trong thiết kế HCI

TYPE OF GUIDANCE

Specifications

Principles and general recommendations

APPLICATION AREA

General guidance

Equipment

Environment

Software

User Testing

Process

SCOPE: phần này cung cấp các khuyến nghị cho các hệ thống sử dụng các giao diện dạng form điện. Với trong tác form điện, người dùng nhìn thấy một màn hình với các trường có liên quan và nhập dữ liệu vào chỗ yêu cầu. Chuẩn này trước tiên mô tả các miền ứng dụng tương ứng cho các giao diện này (khi người dùng đã quen với form giấy nhưng chưa quen với máy tính). Các khuyến nghị bao gồm: cấu trúc form điện, khuyến cáo đầu vào, phản hồi và dịch chuyển. Chuẩn chứa 2 phụ lục. Đầu tiên là một thủ tục mẫu để đánh giá khả năng áp dụng được và tôn trọng và bao gồm 9-page checklist. Phụ lục hai là một thư viện và các bảng tham khảo chéo tới các khuyến nghị trong chuẩn tới các nguồn lý thuyết.

AUDIENCE: Cho người thiết kế và đánh giá giao diện người dùng.

III. CHUẨN ISO 11581

Chuẩn gồm 6 phần song phụ lục này chỉ trình bày 4 phần: phần 1, phần 2, phần 3 và phần 6.

1. Icons - general

ISO 11581-1 cung cấp bộ khung cho việc phát triển và thiết kế các icons và khả năng ứng dụng của chúng trên màn hình để hiển thị đồ họa cũng như văn bản. Nó bao gồm:

- Các khuyến nghị chung và các khuyến nghị cho icon
- Các biến thể toàn cục cho các biểu diễn đồ họa của icon.

a. Bộ khung khái niệm

Việc tạo ra các biểu tượng cho phép người dùng liên kết hình ảnh của nó tới chức năng mà nó thực hiện. Framework sử dụng để thiết kế và phát triển icon.

Phụ lục C: Các chuẩn trong thiết kế HCI

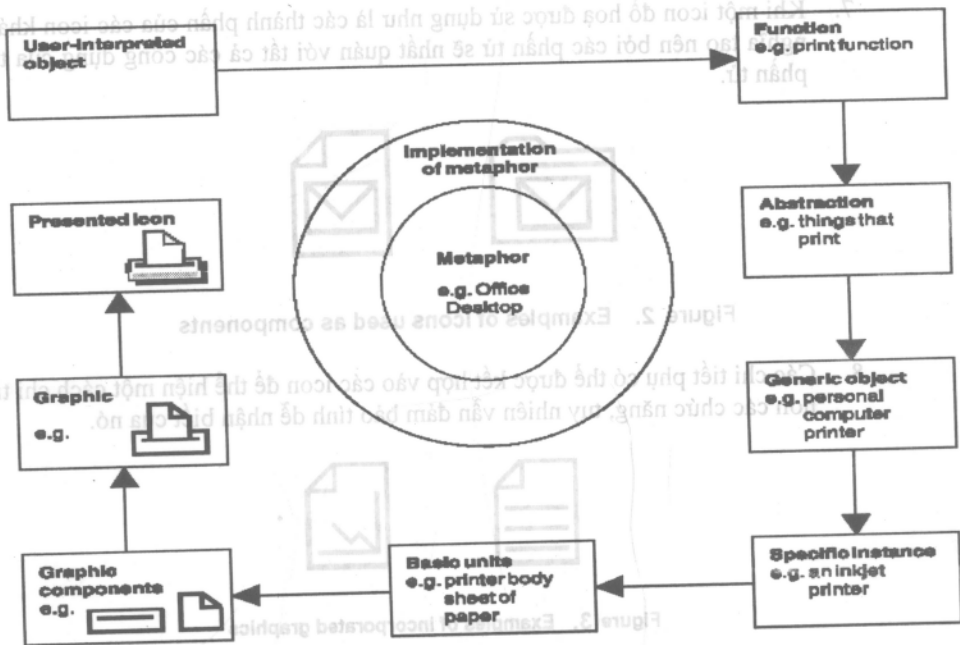


Figure 1. Framework used to specify icons

b. Các yêu cầu và các khuyến nghị cho biểu tượng

• Các yêu cầu:

1. Giao diện của một biểu tượng khi biểu diễn một trạng thái hoặc một chế độ của hệ thống máy tính sẽ được phân biệt rõ ràng với biểu diễn của một trạng thái hoặc chế độ khác
2. Một icon sẽ vẫn dễ hiểu và dễ phân biệt khi có bất kỳ sự thay đổi nào về giao diện do sự thay đổi về trạng thái hoặc chế độ, trong môi trường mà nó được sử dụng.
3. Tất cả các biểu tượng sẽ tuân theo mệnh đề 4, 5 trong ISO 9241-3:1992
4. Bất cứ khi nào một biểu tượng được di chuyển từ một biểu tượng khác, nhưng không phải là kích hoạt bất kỳ một vùng nhạy cảm nào, thì vùng nhạy cảm xếp chồng của icon dịch chuyển sẽ ở trên các icon khác.
5. Sự tương tác với các icon sẽ không xóa đi bất cứ một dữ liệu nào mà không được phép của người dùng.
6. Màu sắc sẽ không được sử dụng như là các phần tử thông tin duy nhất để phân biệt giữa các icon trừ khi biểu diễn của phần tử chức năng chính là màu đó.

7. Khi một icon đồ họa được sử dụng như là các thành phần của các icon khác, ý nghĩa tạo nên bởi các phần tử sẽ nhất quán với tất cả các công dụng của từng phần tử.



Figure 2. Examples of Icons used as components

8. Các chi tiết phụ có thể được kết hợp vào các icon để thể hiện một cách chi tiết hơn các chức năng, tuy nhiên vẫn đảm bảo tính dễ nhận biết của nó.



Figure 3. Examples of incorporated graphics

- **Các khuyến nghị:**

1. Giao diện của các icon nên nhất quán trong một tập các icon, nghĩa là trong một tập các icon nên được hiển thị cùng một kiểu đồ họa giống nhau.
2. Nếu các icon được biểu diễn ở các kích cỡ khác nhau trên màn hình, thì thiết kế của icon nên vẫn đảm bảo được tính dễ hiểu và dễ nhìn, đảm bảo các thành phần chính của nó vẫn xuất hiện.
3. Nếu các icon được sử dụng trên các màn hình khác nhau làm cho các icon được hiển thị với những tỷ lệ 2 cạnh khác nhau, thì các phương pháp thiết kế nên quan tâm đến việc tạo ra giao diện của icon sao cho nó luôn tương tự như hình ảnh thiết kế ban đầu.
4. Tất cả các icon đã có sẵn nên dễ hiểu. Khi tính dễ hiểu từ lần quan sát đầu tiên không phải là một yêu cầu về tính dùng được, thì các icon nên có khả năng học và dễ nhận biết.
5. Vị trí của các nhãn liên kết với icon có thể thay đổi bởi người dùng nên nhất quán bên trong một môi trường hoặc một tập các môi trường được thiết kế để sử dụng cùng nhau.
6. Hoạt hình không được làm giảm tính dễ hiểu và dễ nhận biết của một icon (mục 5 of ISO 9241-3:1992).

• **Các biến thể toàn cục:**

1. Các biến thể toàn cục cho các thuộc tính đường thẳng như: kiểu, độ rộng, điểm kết thúc, liên thông, mẫu, và màu.

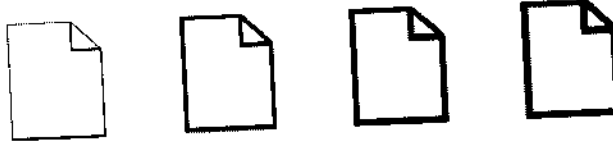


Figure 4. Examples of variations in the line attribute "width".

2. Các biến thể toàn cục cho các thuộc tính cạnh góc là: cong, vuông, và liên thông



Figure 5. Examples of corner attributes

3. Mức độ chi tiết có thể được tăng lên để thêm vào tính thực tế.

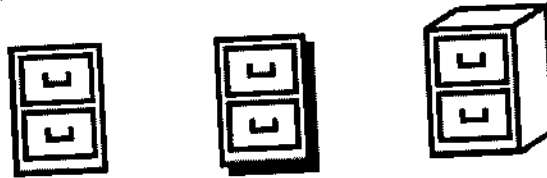


Figure 6. Examples of levels of detail

4. Bề mặt các mẫu hoặc màu sắc có thể khác nhau toàn bộ nhưng không làm giảm tính dễ hiểu của các icon.
5. Các phần tử đồ họa được thêm vào không nên làm giảm tính dễ nhận dạng của icon.

2. Object Icons

Biểu tượng đối tượng - Object icon là một tập con của các icon biểu diễn các đối tượng, tạo nên một miền của một hệ thống hoặc một ứng dụng và người dùng thao tác lên chúng trong khi thực hiện công việc của họ. Chúng được phân biệt với các icon khác vì thực tế chúng dàn xếp các tương tác của người dùng với phần mềm ứng dụng, và chúng có thể được di chuyển, được mở ra. Các biểu tượng đối tượng là những hình ảnh biểu diễn các chức năng bằng cách sử dụng mối liên quan giữa các đối tượng vật lý giống nhau.

Phụ lục C: Các chuẩn trong thiết kế HCI

Bên cạnh việc biểu diễn một chức năng, một biểu tượng đối tượng có thể biểu diễn một đối tượng vật lý tương tự. Ví dụ, một ảnh biểu tượng máy in có thể biểu diễn một máy in cụ thể hơn là khái niệm chung của việc in ấn.

a. Các yêu cầu và các khuyến nghị

- Các yêu cầu của biểu tượng đối tượng:
 1. Biểu diễn và chức năng: nếu một hệ thống hoặc một ứng dụng sử dụng một biểu tượng đối tượng có giao diện được xác định như trong phần đặc tả icon, trong các biến thể cụ thể cho trước và trong các biến thể toàn cục xác định trong ISO/IEC 11581-1, thì nó sẽ có nhiệm vụ như là một chức năng chính xác định trong phần đặc tả icon.
 2. Hướng: Ý nghĩa của một biểu tượng đối tượng có thể phụ thuộc vào hướng của nó, và ta nên cẩn thận để tránh sự nhầm lẫn.
 3. Opacity - độ mờ đục: vùng diện tích bên trong viền của một icon đồ họa là đục. Phần diện tích còn lại bên ngoài đường bao và bị giới hạn bởi viền ôvan có thể là đục hoặc trong suốt.
 4. Mờ một biểu tượng đối tượng: cho phép truy cập tới các chức năng có liên quan hoặc các thông tin của một icon và cung cấp một khung nhìn của đối tượng được biểu diễn bởi icon.
 5. Thay đổi trạng thái: Các quy ước được sử dụng để chỉ ra các thay đổi về trạng thái của các biểu tượng đối tượng sẽ được sử dụng nhất quán cho tất cả các biểu tượng đối tượng.



Figure 7. Examples of state indication

6. Vùng nhạy cảm: Mỗi biểu tượng đối tượng đều có một vùng nhạy cảm. Mục đích của nó là cung cấp một vùng, liên kết với mỗi một icon, cho phép người dùng thao tác và tương tác với các icon khác.
- Các khuyến nghị:
 1. Biểu diễn và chức năng.
 2. Hành vi nhất quán.
 3. Cảnh trí.
 4. Kiểu dạng.
 5. Tính quan sát được.
 6. thiết kế các biểu tượng đối tượng.

Phụ lục C: Các chuẩn trong thiết kế HCI

b. Các đặc tả icon

Có ba dạng đặc tả của object icons

- Biểu tượng đối tượng khung: có thể chấp nhận các đối tượng khác như là đầu vào.
- Biểu tượng đối tượng thứ cấp: không thể chấp nhận các đối tượng khác là đầu vào. Có hai dạng:

- + Biểu tượng đối tượng phụ.
- + Biểu tượng đối tượng môi trường hệ thống.

• Biểu tượng đối tượng khung:

1. Văn bản

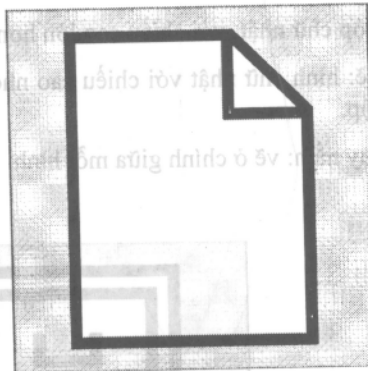
- Chức năng chính: chứa thông tin có thể được biểu diễn tới người dùng, và cung cấp đơn vị lưu trữ thông tin nhỏ nhất cho người dùng, trừ khi ứng dụng yêu cầu văn bản đó chứa các văn bản khác.

- Thể hiện cụ thể: Một trang cụ thể trong một cơ quan

- Các thành phần:

- Một trang giấy. Hình chữ nhật với chiều cao lớn hơn chiều rộng.
- Một góc bị khoét lõm xuống. Hình tam giác bên trong trang giấy.

- Hình



2. Thư mục (Folder):

- Chức năng chính: dùng để chứa các đối tượng như: các tài liệu, các thư mục khác, và các ứng dụng nhưng không phải là các tủ đựng hồ sơ (filing cabinet.)

- Thể hiện cụ thể: Một thùng chứa - a manila folder

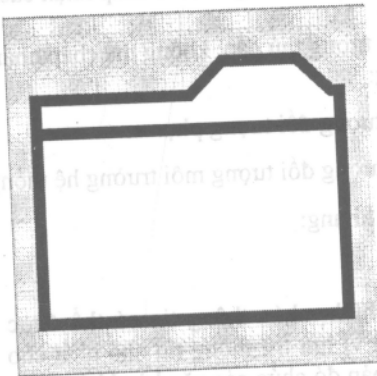
- Các thành phần:

- Mặt sau: hình chữ nhật mà chiều cao nhỏ hơn chiều rộng và có một đường kẻ dọc theo đỉnh.

Phụ lục C: Các chuẩn trong thiết kế HCI

- Mặt trước: hình chữ nhật với chiều cao nhỏ hơn chiều rộng và định vị ở phía trước đối với mặt sau.

- Hình



3. Tủ đựng hồ sơ (Filing cabinet):

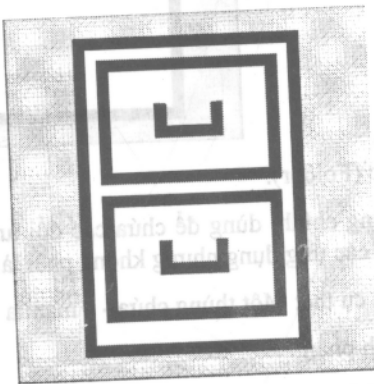
- Chức năng chính: chứa các đối tượng khác như tài liệu, thư mục và các tủ hồ sơ khác, và cung cấp một mức trong cây phân cấp lưu trữ cao hơn thư mục.

- Thể hiện cụ thể: Tủ đựng hồ sơ với 2 ngăn kéo.

- Các thành phần:

- Hộp chữ nhật, với chiều cao lớn hơn chiều rộng.
- Vẽ: hình chữ nhật với chiều cao nhỏ hơn chiều rộng được chồng lên hộp.
- Tay nắm: vẽ ở chính giữa mỗi hình.

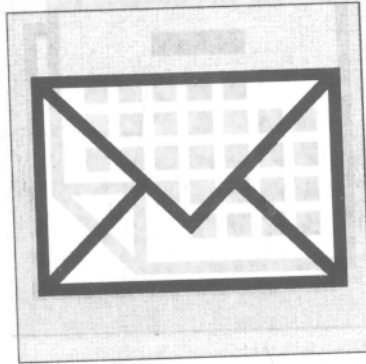
- Hình



Phụ lục C: Các chuẩn trong thiết kế HCI

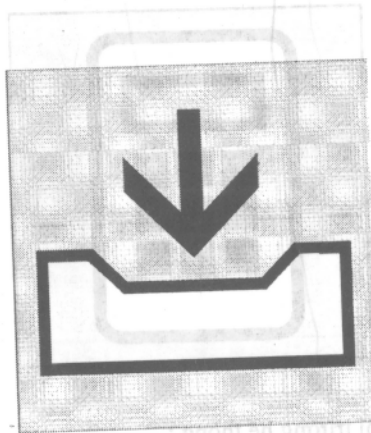
4. Phong bì

- Hình



5. Tủ đựng công văn đến (In-tray)

- Hình

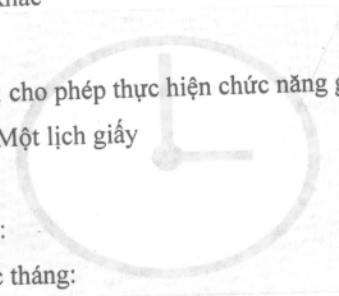


Ngoài ra còn có nhiều các đối tượng khác như tủ đựng công đến, máy in, điện thoại,...

- Các biểu tượng đối tượng khác

1. Lịch

- Chức năng chính: cho phép thực hiện chức năng ghi và lập lịch các sự kiện
- Thể hiện cụ thể: Một lịch giấy
- Các thành phần:
 - Khung lịch:
 - Chi tiết các tháng:

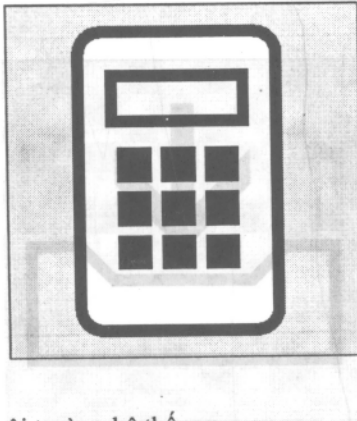


- Hình



2. Máy tính bỏ túi

- Hình



- Biểu tượng đối tượng môi trường hệ thống :

I. Đồng hồ

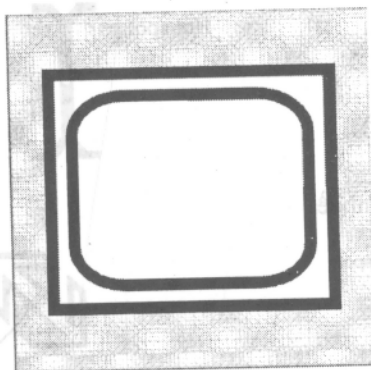
- Hình



Phụ lục C: Các chuẩn trong thiết kế HCI

2. Thiết bị hiển thị

- o Hình



3. Ngoài ra còn có bàn phím, chuột, mạng, thiết bị Audio

3. Pointer icons

- a. Điểm nóng - Hot spot: Các biểu tượng con trỏ thường có một hot spot. Hot spot thông thường là một pixel.

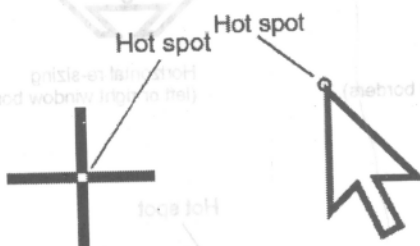
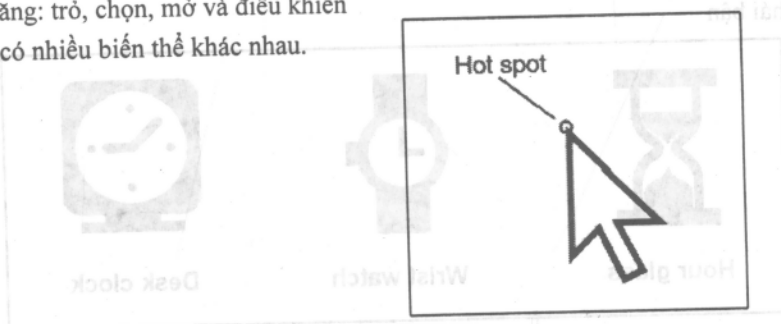


Figure 8. Indication of hot spots

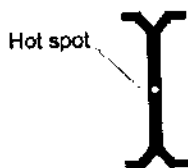
- b. Dạng ngầm định của biểu tượng con trỏ

Chức năng: trỏ, chọn, mở và điều khiển

Chúng có nhiều biến thể khác nhau.



c. Con trỏ định vị văn bản:



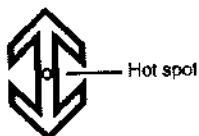
d. Con trỏ biểu thị chỉnh lề



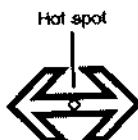
Horizontal and vertical re-sizing
(upper left and lower right window corners).



Horizontal and vertical re-sizing
(upper right and lower left window corners).

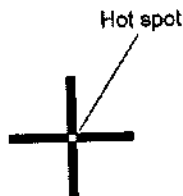


Vertical re-sizing
(upper or lower window borders).



Horizontal re-sizing
(left or right window borders).

e. Con trỏ dạng chữ thập



f. Chỉ trạng thái bận

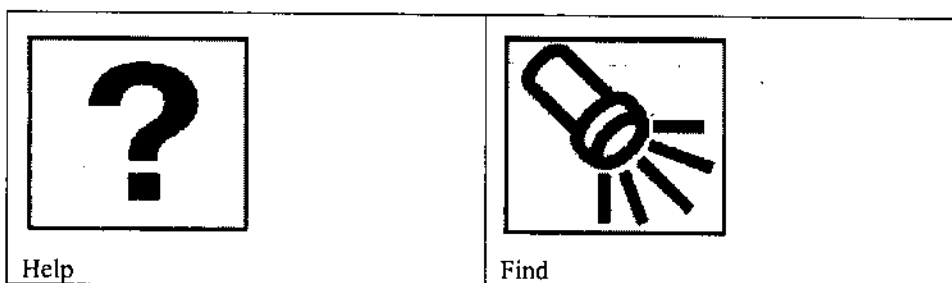


4. Biểu tượng hành động - Action Icons

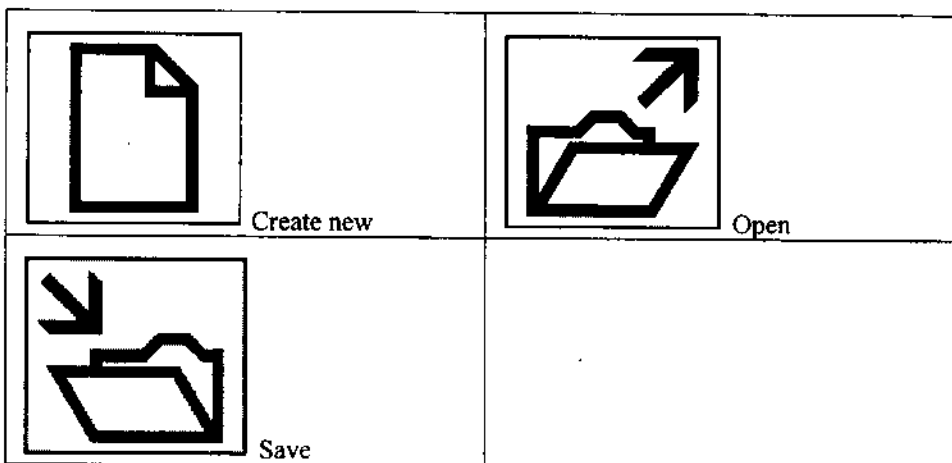
Biểu tượng hành động được phân thành sáu nhóm:

- Hành động chung.
- Hành động điện.
- Hành động in.
- Hành động cho các thao tác soạn thảo.
- Hủy bỏ hay khôi phục hành động.
- Hành động với văn bản.

a. Hành động chung

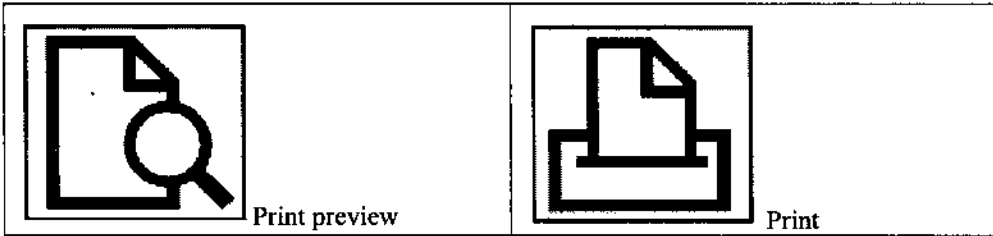


b. Hành động điện

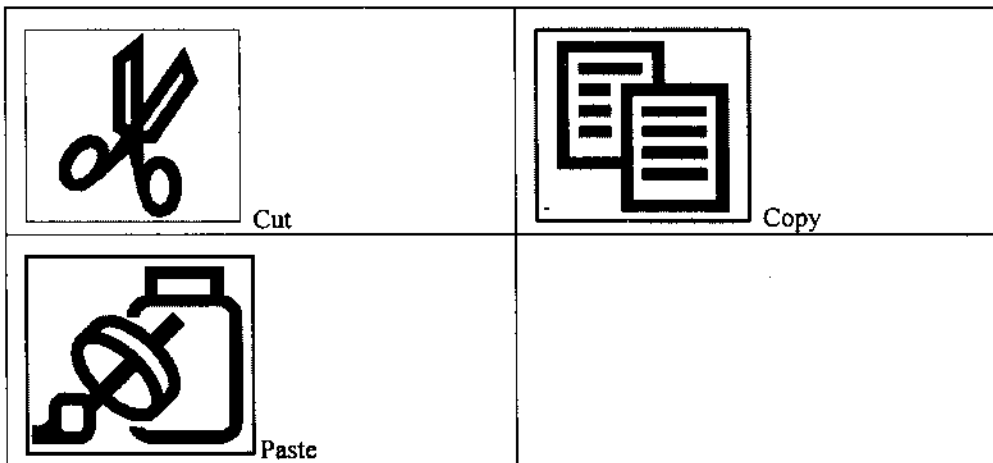


Phụ lục C: Các chuẩn trong thiết kế HCI

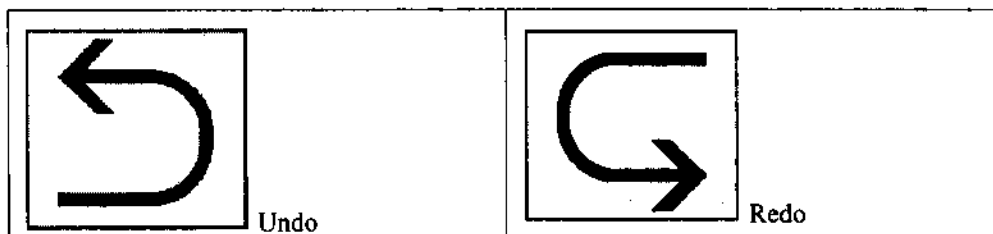
c. Hành động in



d. Hành động cơ bản của soạn thảo


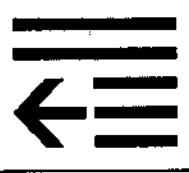

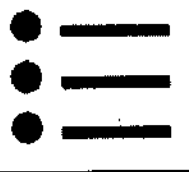





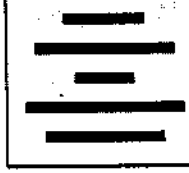
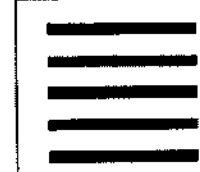


e. Hủy bỏ hay khôi phục hành động



Phụ lục C: Các chuẩn trong thiết kế HCI

f. Hành động với văn bản

	
<p>Increase indent</p>	<p>Decrease indent</p>
	
<p>Enumerate/numbering</p>	<p>Itemise</p>
	
<p>Make Bold</p>	<p>Italicise</p>
	
<p>Underline</p>	<p>Align left</p>
	
<p>Align right</p>	<p>Centralize</p>
	
<p>Full justify</p>	

TÀI LIỆU THAM KHẢO

- [1] *Jenny Preece et al: Human-Computer Interaction*. Addison-Wesley, 1994.
- [2] *Akain J. Dix, Finlay, Abowd and Beale: Human-Computer Interaction (course and book)*. Prentice-Hall, second edition, 1993.
- [3] *Gary Perlman, Ohio State University: User Interface development (course)* SEI Curriculum Module, 11-1989.
- [4] *C. Machgeels: Interfaces Humains-Ordinateur (cours)* Département Informatique, ULB, 2001-2002.
- [5] *J. Coutas, Interface Home-ordinateur, Conception et réalisation*, Dunod Paris 1990.
- [6] *Andy Downton, Engineering the Human-Computer Interface*, McGraw Hill 1992.
- [7] *J. P. Menadier, l'Interface utilisateur pour une Informatique plus conviviale*, Dunod Paris, 1991.
- [8] *Sommerville, Software Engineering (chapter 17: User Interface)*, Addison 1996.
- [9] *PGS. TS. Nguyễn Văn Ba, Bản dự thảo đề cương môn học HCI*, Khoa CNTT Đại học Bách khoa Hà Nội, 1997.
- [10] *Hiệp hội CNPM ACM & IEEE, Bản đề cương HCI (Human Computer Interface - course CS1 (Use Interface design and Development), Chapter 3, page 7-8/14*, New York, 1992.
- [11] *Lương Mạnh Bá, Huỳnh Quyết Thắng, Lê Tấn Hùng, Đề cương chi tiết về HCI*, Hà Nội, 12/2001.
- [12] *Lương Mạnh Bá, Slides bài giảng về HCI cho sinh viên ngành CNPM, ĐHBK Hà Nội*, 09/2002.
- [13] *Standards and Guidelines*, <ftp:advice.cis.ohio.state.edu/pub/hci/guidelines>
- [14] *Nivel Bevan, Human-Computer Interaction Standards, Proceedings of the 6th International Conference on Human-Computer Interaction, Yokohama, July 1995*, Anzai & Ogawa, Elsevier
- [15] *Jacob Nielsen, Designing Web Usability*, 2000
- [16] *J. Patrick Lynch & Sarah Horton, Web Style Guide* (<http://www.info.med.yale.edu/caim/manual>)
- [17] *S. Riihiaho, Experiences with usability evaluation methods, Licentiate's thesis*, Helsinki University of Technology, 1998.

- [18] *Lương Mạnh Bá, Huỳnh Quyết Thắng, Lê Tấn Hùng*, Tương tác người - máy. Ứng dụng kỹ thuật đồ họa và công nghệ tác tử trong thiết kế giao tiếp người dùng. Đề tài cấp Nhà nước KC01-09. Báo cáo tại hội thảo khoa học - đề tài KC01-09 ĐHBK Hà Nội, Hà Nội 12/2002.
- [19] *Lương Mạnh Bá*, Một số phương pháp đánh giá tính tiện dụng trong thiết kế tương tác người - máy, báo cáo tại Hội nghị ICT-RDA'03, Hà Nội 3-2003 và đăng trong kỷ yếu của Hội nghị.
- [20] [Webdesign.about.com/library](http://webdesign.about.com/library).
- [21] [Usable web.com/authors/](http://Usableweb.com/authors/)
- [22] [http:// www.cis.ohio-state.edu/hypertext/faq/usenet](http://www.cis.ohio-state.edu/hypertext/faq/usenet).
- [23] Steve Krug - Don't Make me Think, 2001.

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN**

**TƯƠNG TÁC NGƯỜI - MÁY
(HUMAN - COMPUTER INTERACTION)**

Tác giả: Lương Mạnh Bá

Chịu trách nhiệm xuất bản: PGS. TS. TÔ ĐĂNG HẢI
Biên tập và sửa bài: ThS. NGUYỄN HUY TIẾN
Trình bày bìa: NGỌC LINH
HƯƠNG LAN

**NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT
70 Trần Hưng Đạo - Hà Nội**

In 800 bản, khổ 16 x 24 cm tại Xí nghiệp in NXB Lý luận chính trị
Giấy phép xuất bản số: 546- 12, cấp ngày 23/8/2005
In xong và nộp lưu chiểu tháng 11 năm 2005.

205255



Giá: 38.000đ



PHẦN I: TỔNG QUAN VỀ CÁC HỆ TƯƠNG TÁC

Chương I. Giới thiệu chung

Chương II. Các yếu tố then chốt trong tương tác người máy


Chương III. Các dạng tương tác giữa con người với máy tính

Chương IV. Giao diện trong các hệ tương tác



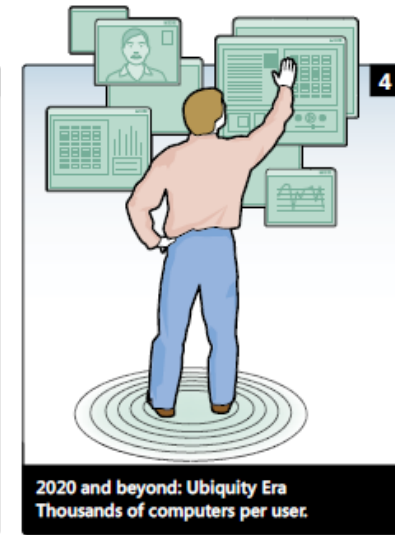
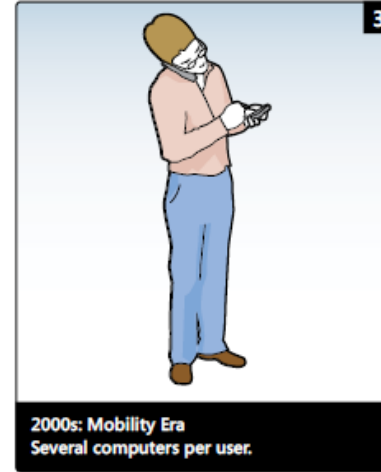
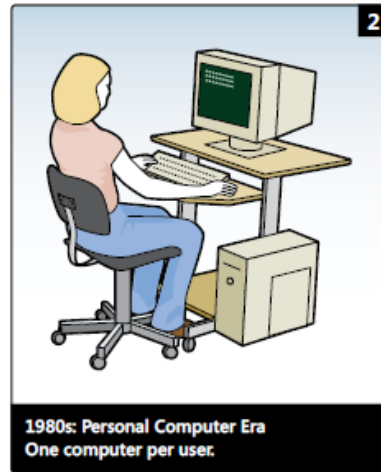
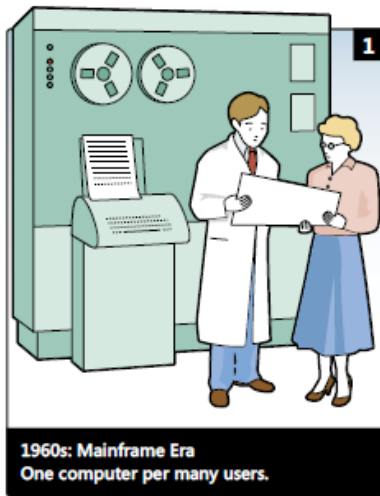


CHƯƠNG I: GIỚI THIỆU CHUNG

- 
- I. Định nghĩa**
 - II. Tính dùng được
 - III. Cách tổ chức các hệ tương tác
 - IV. Ví dụ

1. Hệ tương tác

- Hệ thống bất kỳ, chấp nhận đầu vào từ người sử dụng và cung cấp thông tin như đầu ra cho người sử dụng.



1961

1981

2111

2121

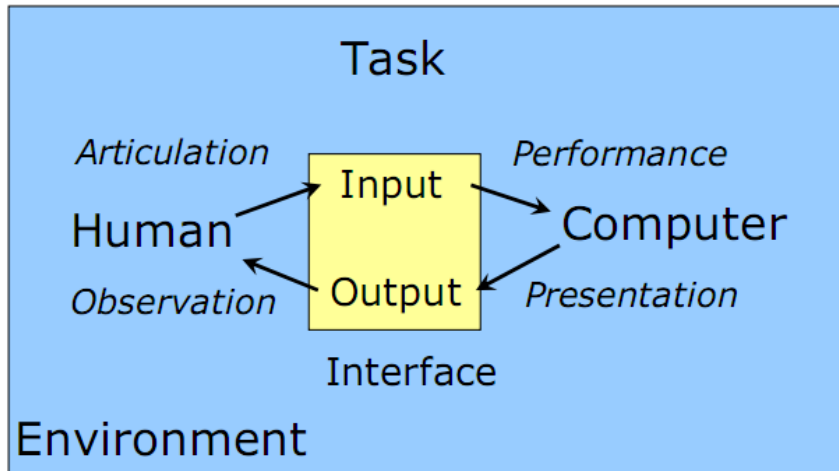




Ví dụ

- Đây là hệ tương tác
 - Đèn bàn
 - Lò vi sóng
 - Điện thoại di động
 - Máy tính cầm tay
 - Máy bay chở khách
- Các ví dụ khác ?

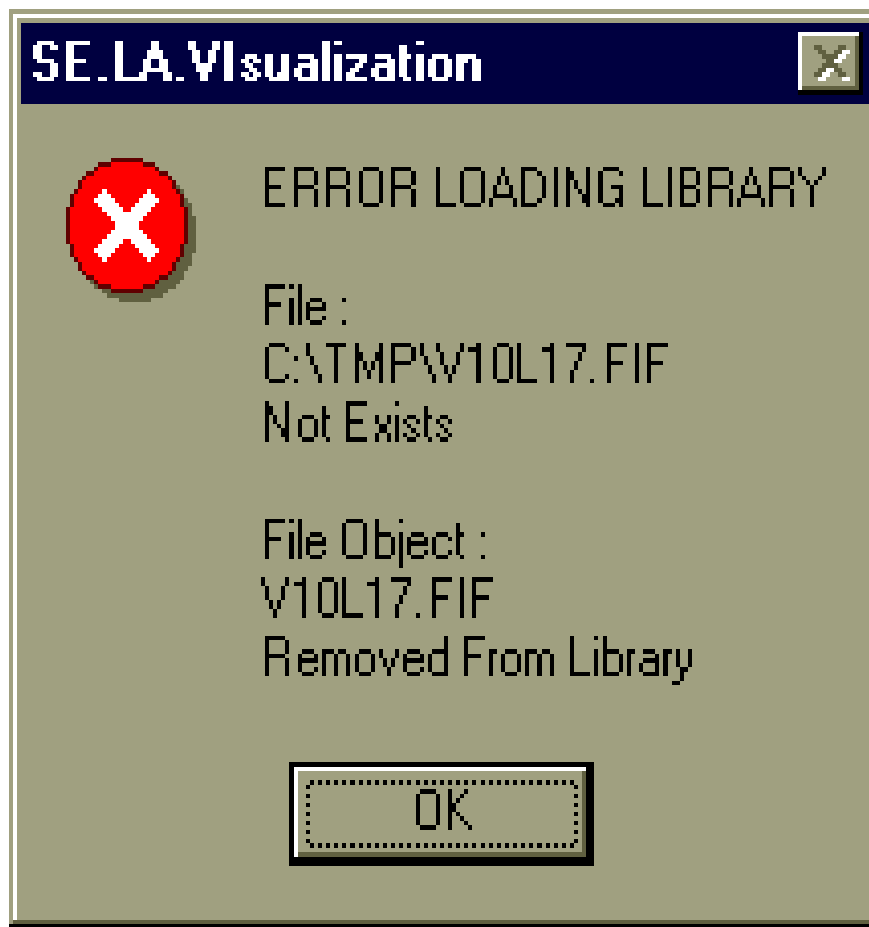
2. Tương tác người-máy tính



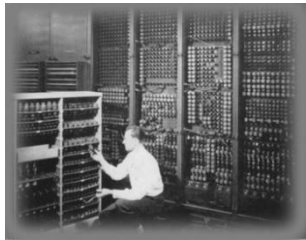
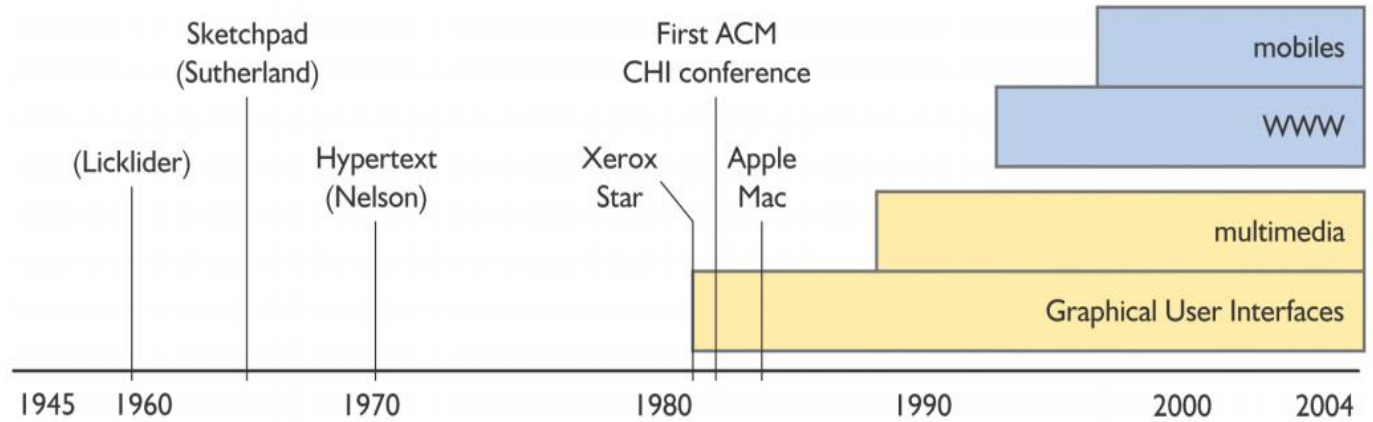
- Baecker & Buxton, 1987: Tập các quá trình, đối thoại và các hành động, qua đó con người sử dụng và tương tác với máy tính.

- HCI: Human - Computer Interaction
- CHI: Computer - Human Interaction
- IHO: Interaction Humains Ordinateur
- IHM: Interaction Homme Machine

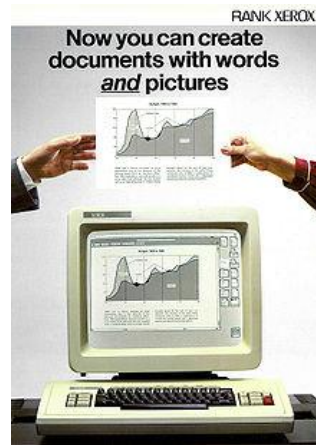
Ví dụ



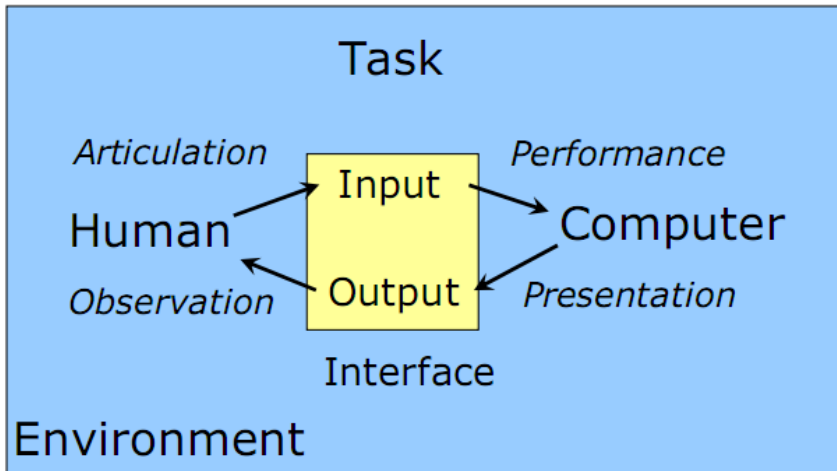
2. Tương tác người-máy tính



Ivan Sutherland à la console du TX-2, travaillant sur SketchPad (MIT, 1963).
Source : Biography of Ivan Sutherland



2. Tương tác người-máy tính



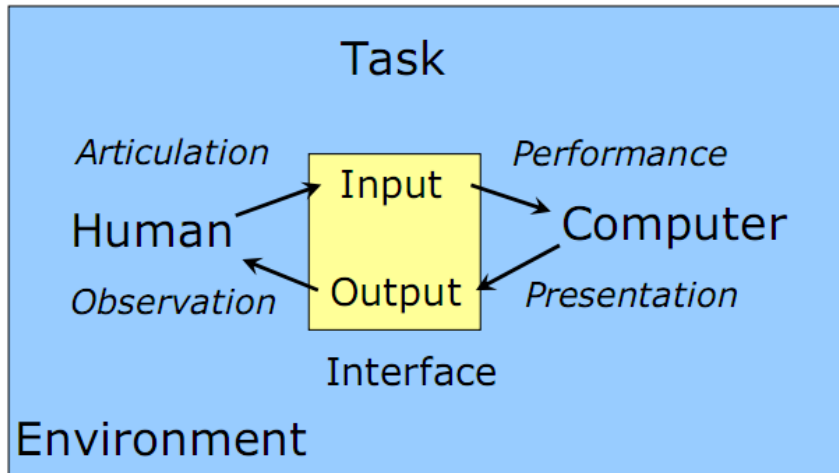
- HCI: Human - Computer Interaction
- CHI: Computer - Human Interaction
- IHO: Interaction Humains Ordinateur
- IHM: Interaction Homme Machine

- Baecker & Buxton, 1987: Tập các quá trình, đối thoại và các hành động, qua đó con người sử dụng và tương tác với máy tính.
- ACM SIGCHI 1992: Là một lĩnh vực liên quan đến thiết kế, đánh giá và cài đặt hệ thống máy tính tương tác cho con người sử dụng và nghiên cứu các hiện tượng chính xảy ra trên đó.

Ví dụ



2. Tương tác người-máy tính

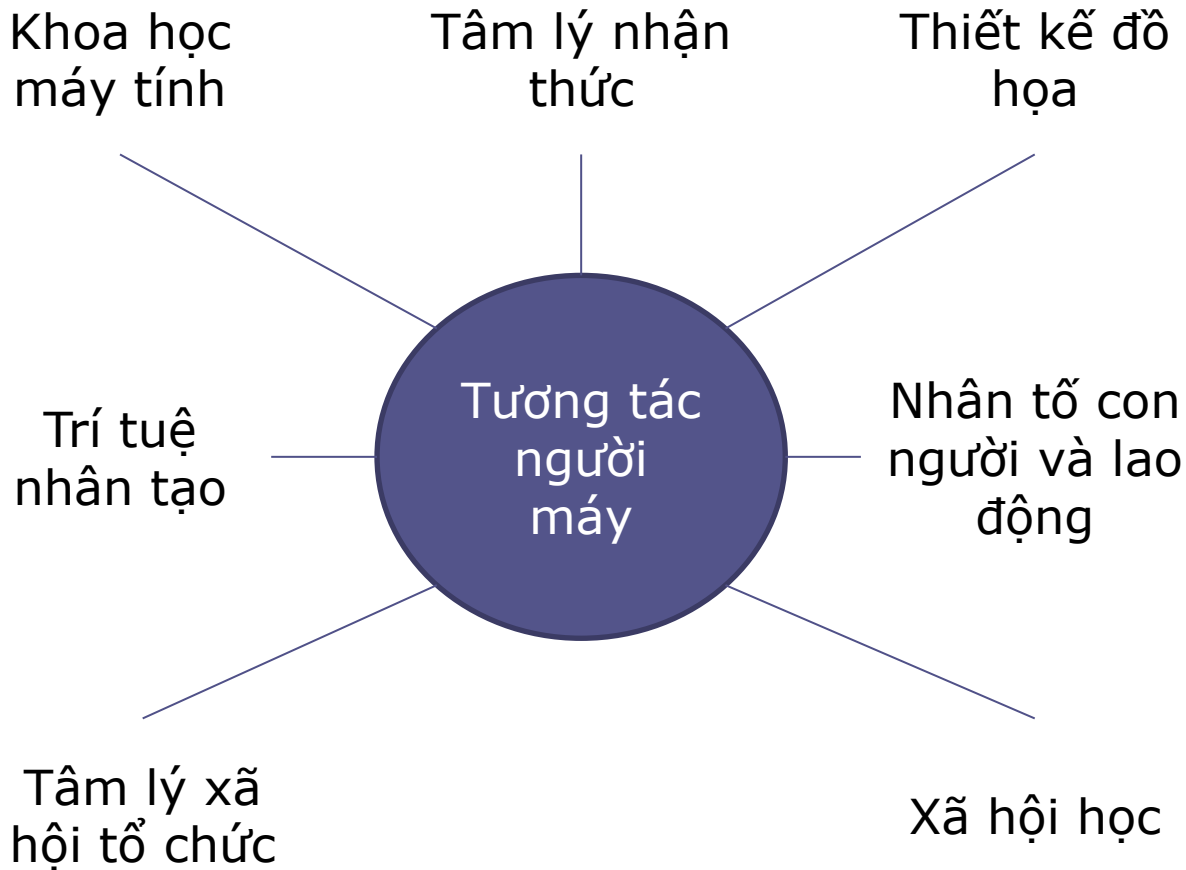


- HCI: Human - Computer Interaction
- CHI: Computer - Human Interaction
- IHO: Interaction Humains Ordinateur
- IHM: Interaction Homme Machine

- Backer & Buxton, 1987: Tập các quá trình, đối thoại và các hành động, qua đó con người sử dụng và tương tác với máy tính.
- ACM SIGCHI 1992: Là một lĩnh vực liên quan đến thiết kế, đánh giá và cài đặt hệ thống máy tính tương tác cho con người sử dụng và nghiên cứu các hiện tượng chính xảy ra trên đó.

Tại sao HCI lại quan trọng ?

Vị trí, vai trò của HCI



Tạo ra các hệ thống cung cấp các chức năng an toàn và tiện dụng (usability)



CHƯƠNG I: GIỚI THIỆU CHUNG



I. Định nghĩa

II. Tính dùng được

III. Cách tổ chức các hệ tương tác

IV. Ví dụ

1. Ví dụ: ai chơi trò này

A Bug's Life: A Bug's Land

This official Bug's Life Flash online game is 0.45 MB in size, so please allow some time for it to load...

Block...

Disney Pixar
bug's life
collector's edition
coming **may 27th**

a bug's land

Help Flik in his journey to the city by guiding his dandelion seed and gathering extra nuts and leaves for the colony.
Watch out for rocks and birds!

INSTRUCTIONS

START GAME

send to a friend

Also visit a **bug's land** at California Adventure

© Disney Enterprises, Inc. / Pixar Animation Studios. All Rights Reserved

Tính dùng được: tìm giá phòng đôi tại khách sạn Holiday Inn, Bradley

Pennsylvania

Bedford Motel/Hotel: Crinaline Courts

(814) 623-9511 S: \$18 D: \$20

Bedford Motel/Hotel: Holiday Inn

(814) 623-9006 S: \$29 D: \$36

Bedford Motel/Hotel: Midway

(814) 623-8107 S: \$21 D: \$26

Bedford Motel/Hotel: Penn Manor

(814) 623-8177 S: \$19 D: \$25

Bedford Motel/Hotel: Quality Inn

(814) 623-5189 S: \$23 D: \$28

Bedford Motel/Hotel: Terrace

(814) 623-5111 S: \$22 D: \$24

Bradley Motel/Hotel: De Soto

(814) 362-3567 S: \$20 D: \$24

Bradley Motel/Hotel: Holiday House

(814) 362-4511 S: \$22 D: \$25

Bradley Motel/Hotel: Holiday Inn

(814) 362-4501 S: \$32 D: \$40

Breezewood Motel/Hotel: Best Western Plaza

(814) 735-4352 S: \$20 D: \$27

Breezewood Motel/Hotel: Motel 70

(814) 735-4385 S: \$16 D: \$18

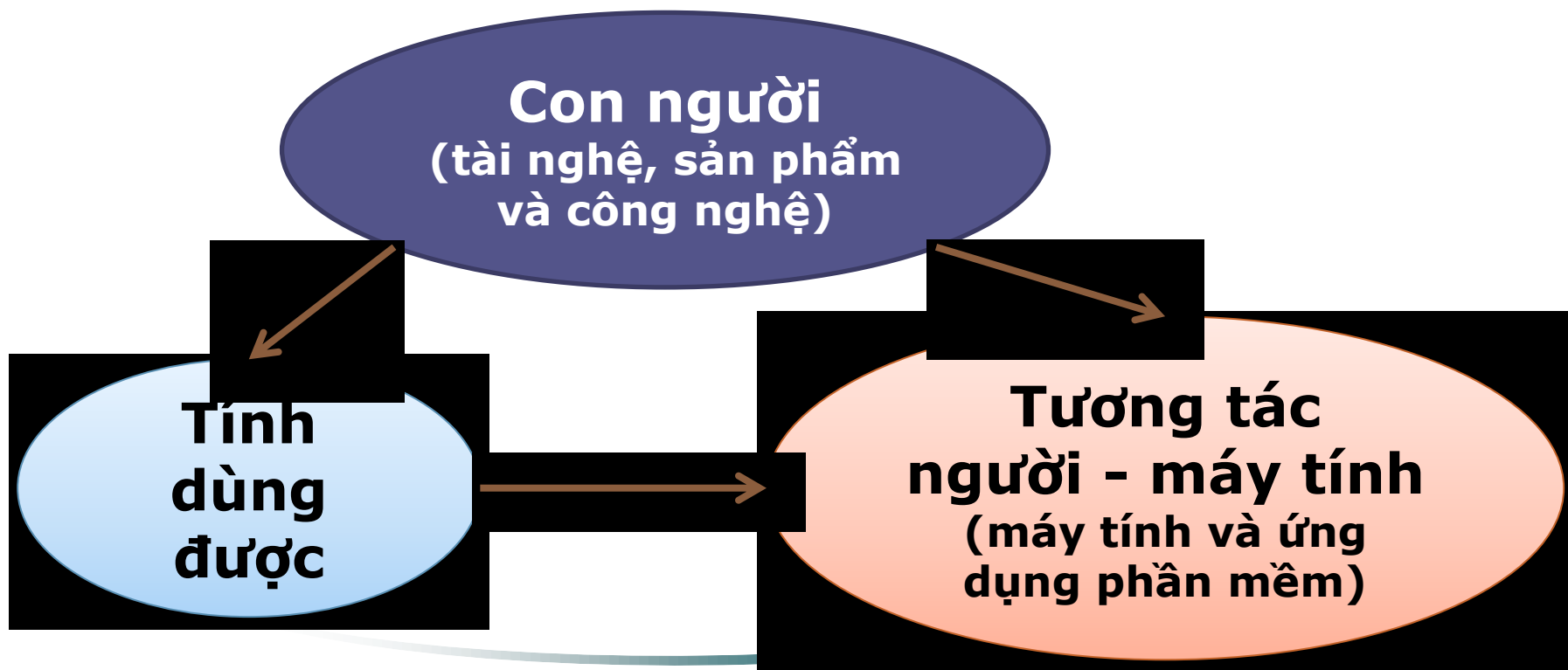
Tính dùng được: tìm giá phòng đôi tại khách sạn Quality Inn, Columbia

South Carolina

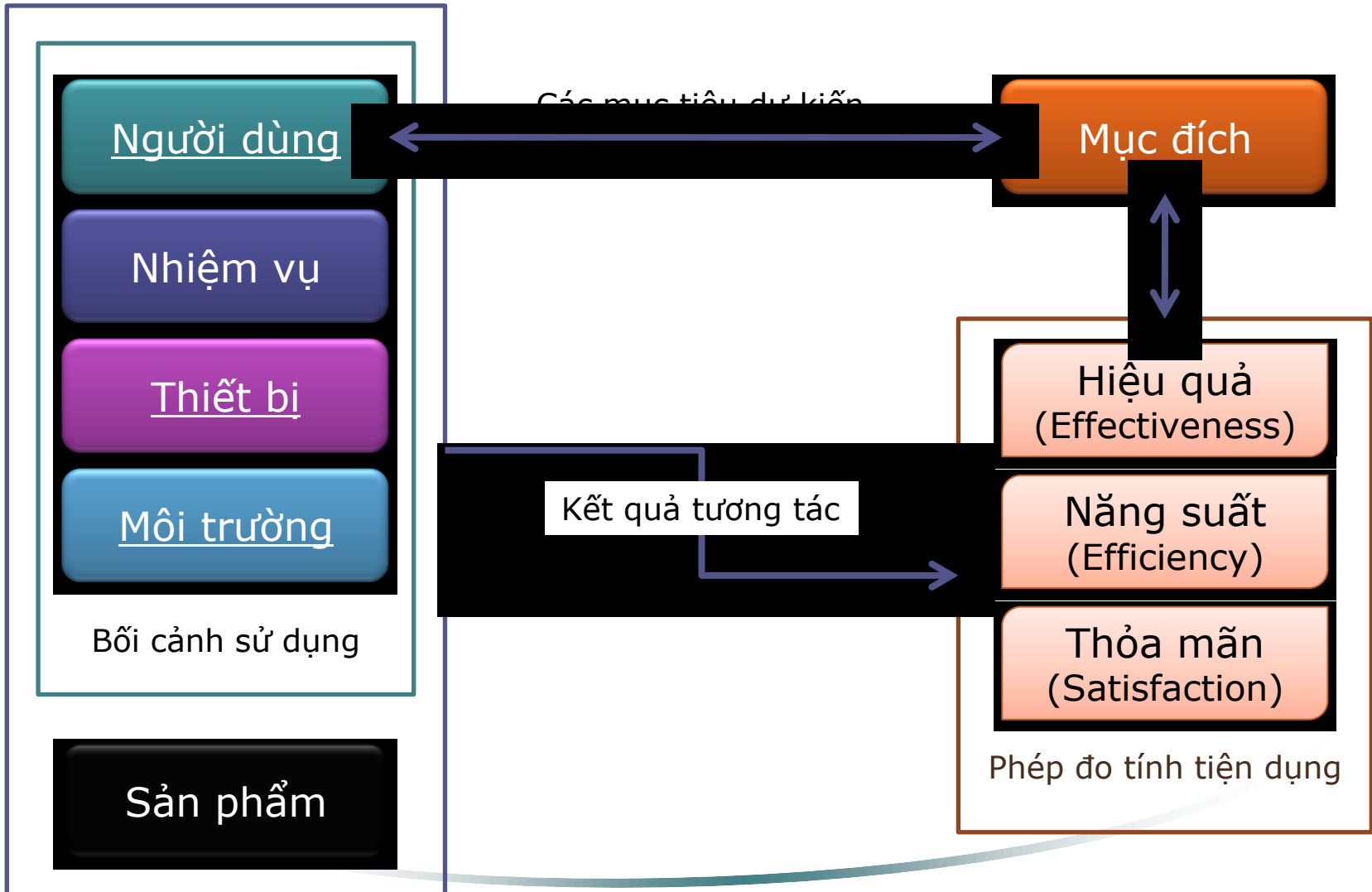
City	Motel/Hotel	Area code	Phone	Rates	
				Single	Double
Charleston	Best Western	803	747-0961	\$26	\$30
Charleston	Days Inn	803	881-1000	\$18	\$24
Charleston	Holiday Inn N	803	744-1621	\$36	\$46
Charleston	Holiday Inn SW	803	556-7100	\$33	\$47
Charleston	Howard Johnsons	803	524-4148	\$31	\$36
Charleston	Ramada Inn	803	774-8281	\$33	\$40
Charleston	Sheraton Inn	803	744-2401	\$34	\$42
Columbia	Best Western	803	796-9400	\$29	\$34
Columbia	Carolina Inn	803	799-8200	\$42	\$48
Columbia	Days Inn	803	736-0000	\$23	\$27
Columbia	Holiday Inn NW	803	794-9440	\$32	\$39
Columbia	Howard Johnsons	803	772-7200	\$25	\$27
Columbia	Quality Inn	803	772-0270	\$34	\$41
Columbia	Ramada Inn	803	796-2700	\$36	\$44
Columbia	Vagabond Inn	803	796-6240	\$27	\$30

2. Định nghĩa tính dùng được

- Làm cho hệ thống dễ học và dễ dùng
- Phụ thuộc vào quá trình thiết kế và cài đặt ứng dụng

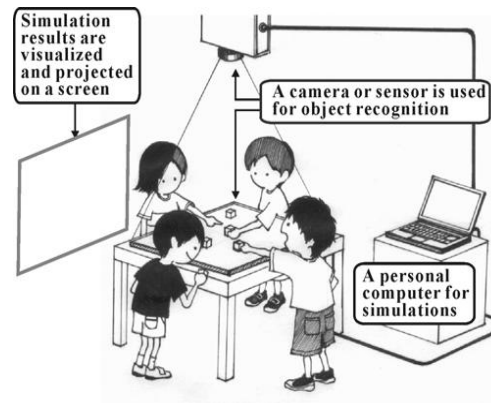
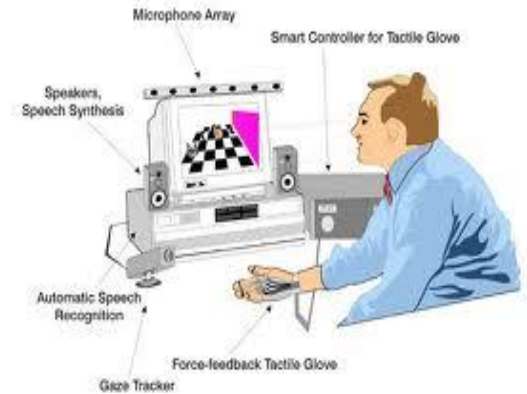


3. Tính dùng được theo ISO 9241-11



a. Người dùng

- Trước đây:
 - Kỹ thuật viên, chuyên gia
- Hiện nay: Đa dạng
 - Người dùng đầu cuối: có ít kiến thức về tin học
 - Kỹ thuật viên, chuyên gia



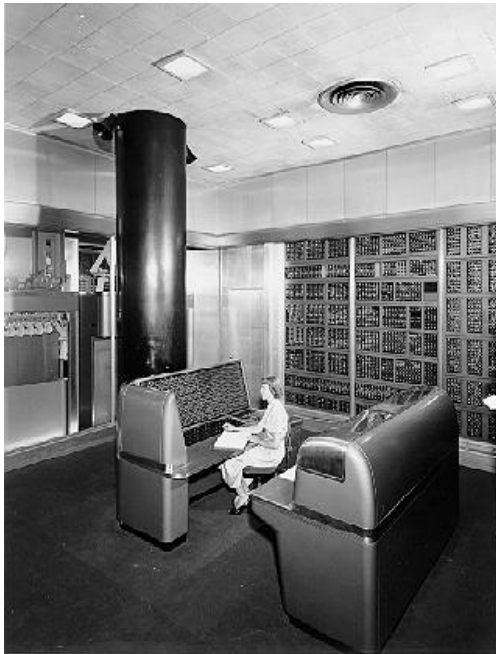
b. Thiết bị

- Âm thanh
- Hình ảnh
- Video
- Đặc điểm
 - Kích thước từ nhỏ đến lớn
 - Di động (PDA, phone)
 - Đàn hồi (Plasticity)
 - Phụ thuộc ngữ cảnh
 - Có thể cá nhân hóa
 - Khắp nơi (Ubiquitous)

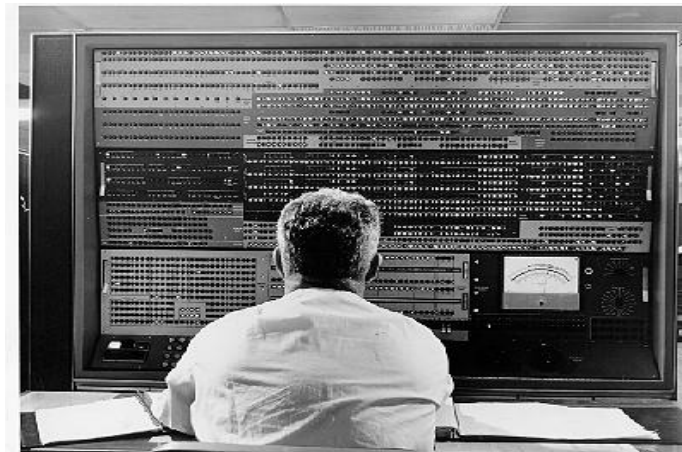


c. Môi trường làm việc

- Trước đây:
 - Máy tính lớn, không nối mạng
 - Người sử dụng máy tính: chuyên gia, kỹ thuật viên
 - Môi trường: văn bản dạng text

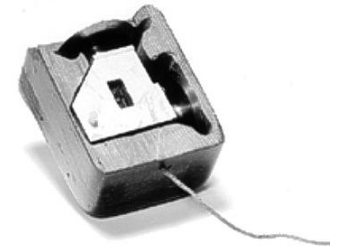


IBM SSEC (1948)



Stretch (1961)

A close-up of the Stretch technical control panel



The First Mouse (1964)
Douglas Engelbart

Môi trường làm việc



IBM 51II, introduced in September 1975, was IBM's first portable computer.

```
TBM12 COM 24973 1-06-94 11:55a STPIPX COM 11515 5-13-94 1:20p
STPUDP COM 13063 5-13-94 1:20p NULL COM 278 2-02-93 4:07p
NE1000 COM 19807 7-30-93 9:40a NE1500T COM 29258 7-30-93 9:41a
NE2 COM 20192 7-30-93 9:36a NE2000 COM 21188 7-30-93 9:38a
NE2100 COM 29256 7-30-93 9:41a NE2_32 COM 19903 7-30-93 9:39a
NE3200 COM 26568 10-20-93 11:28a NTR2000 COM 24909 3-31-93 9:46a
PCIODI COM 26368 5-29-97 7:17p
17 File(s) 113102848 bytes free

C:\>ipxodi

NetWare IPX/SPX Protocol With Mobile Support v3.00 ALPHA 3 (940622)
(C) Copyright 1990-1994 Novell, Inc. All Rights Reserved.

IPXODI-300-13: The LSL is not loaded. Please load the LSL then IPXODI.

C:\>lsl
NetWare Link Support Layer v2.11 BETA 04 (940614)
(C) Copyright 1990-1994 Novell, Inc. All Rights Reserved.

The configuration file used was "C:\NWCLIENT\NET.CFG".
Max Boards 4, Max Stacks 4

C:\>
```

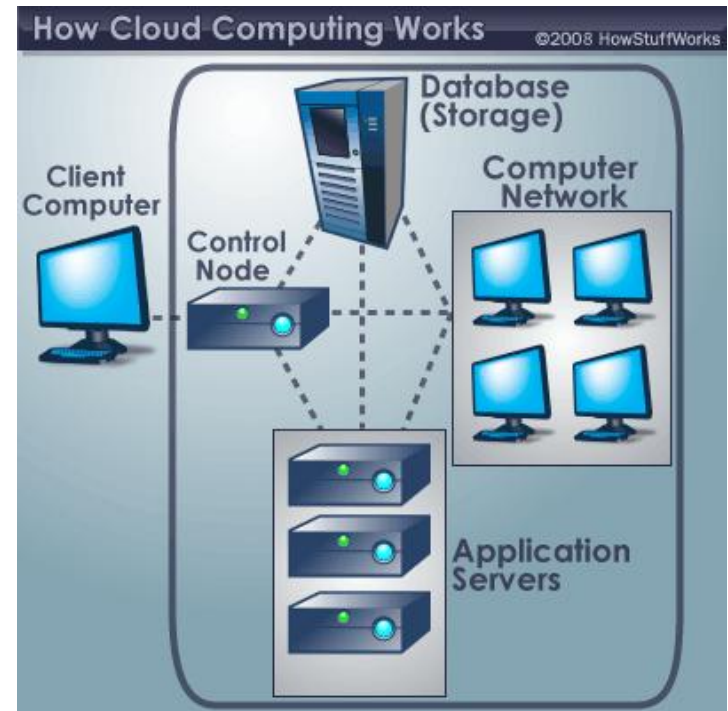
MSDoS – Bill Gate 1981

Môi trường làm việc

- Hiện nay:
 - Máy tính cá nhân, mạng, internet
 - Môi trường: đa dạng, văn bản, đồ họa, trực quan



Giao diện đa phương tiện



Ví dụ về một hệ thống tính toán đám mây


4. Đo tính dùng được theo ISO 9241-11

ISO 9241, *Ergonomics of Human System Interaction*, adopts traditional usability categories with specific measures, e.g.:

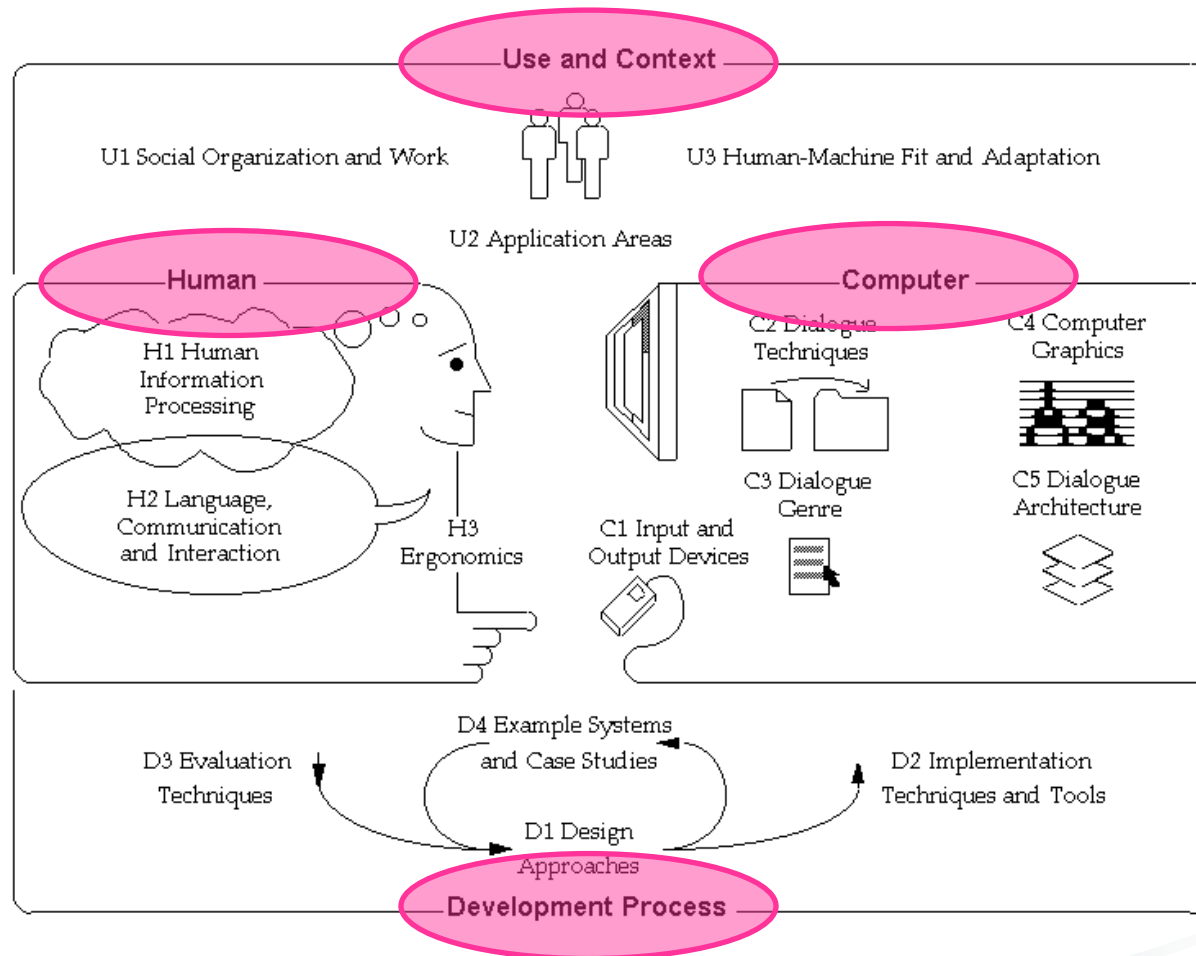
Usability objective	Effectiveness measures	Efficiency measures	Satisfaction measures
Suitability for the task	Percentage of goals achieved	Time to complete a task	Rating scale for satisfaction
Appropriate for trained users	Number of power features used	Efficiency relative to expert user	Rating scale for ease of learning
Learnability	Percentage of functions learned	Time to learn criterion	Rating scale for ease of learning
Error tolerance	Percentage of errors corrected successfully	Time spent on correcting errors	Rating scale for error handling



CHƯƠNG I: GIỚI THIỆU CHUNG

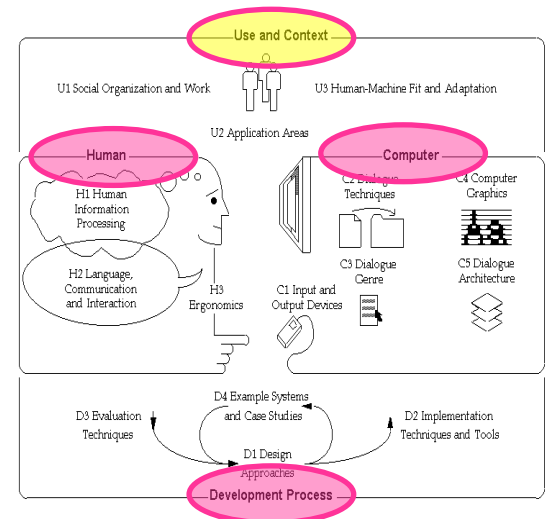
- 
- I. Định nghĩa
 - II. Tính dùng được
 - III. Cách tổ chức các hệ tương tác**
 - IV. Ví dụ

1. Các thành phần chính của hệ tương tác



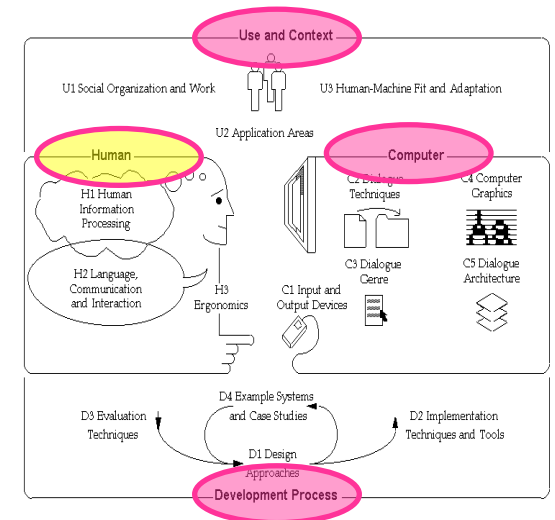
a. Môi trường và ngữ cảnh

- Ứng dụng sử dụng tài nguyên máy tính
- Các tổ chức: xã hội, kinh doanh, v.v tương tác với các ứng dụng đó để hoàn thành một nhiệm vụ đề ra
- Lĩnh vực tương tác:
 - Cá nhân hoặc nhóm
 - Giao tiếp hướng văn bản
 - Giao tiếp hướng thông điệp
 - Trợ giúp trực tuyến hoặc điều khiển hệ thống liên tục
 - Trợ giúp thiết kế, v.v



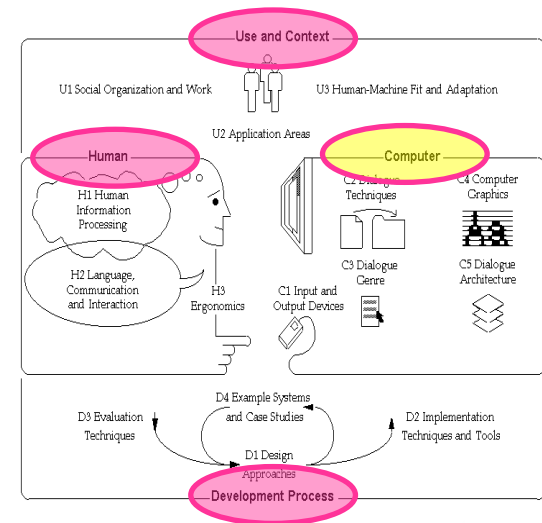
b. Con người

- Mục đích: Hiểu con người như là một Bộ xử lý thông tin
 - Cảm nhận (perception)
 - Lưu trữ (các loại bộ nhớ - Storage)
 - Xử lý (Processing)
- Ngôn ngữ, giao tiếp và tương tác
 - Các sắc thái ngôn ngữ: cú pháp , ngữ nghĩa
 - Các mô hình hình thức của ngôn ngữ
- Công thái học (ergonomie)
 - Bố trí hiện và điều khiển, quan hệ
 - Nhận thức của con người và giới hạn,...



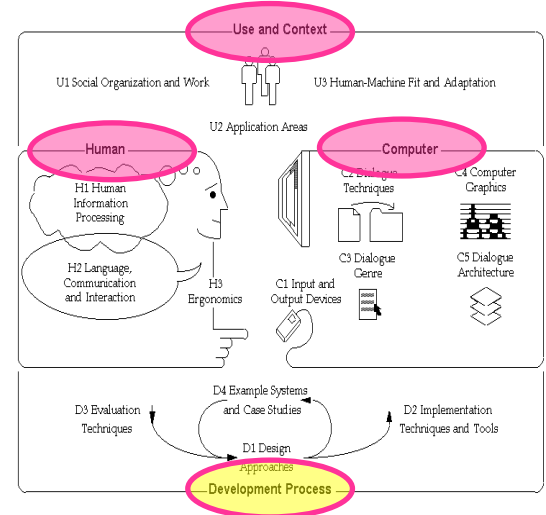
c. Máy tính và kiến trúc tương tác

- Các thiết bị vào ra
- Các kỹ thuật đối thoại; vào, ra và tương tác
- Các kiểu đối thoại
- Đồ họa máy tính
- Kiến trúc đối thoại

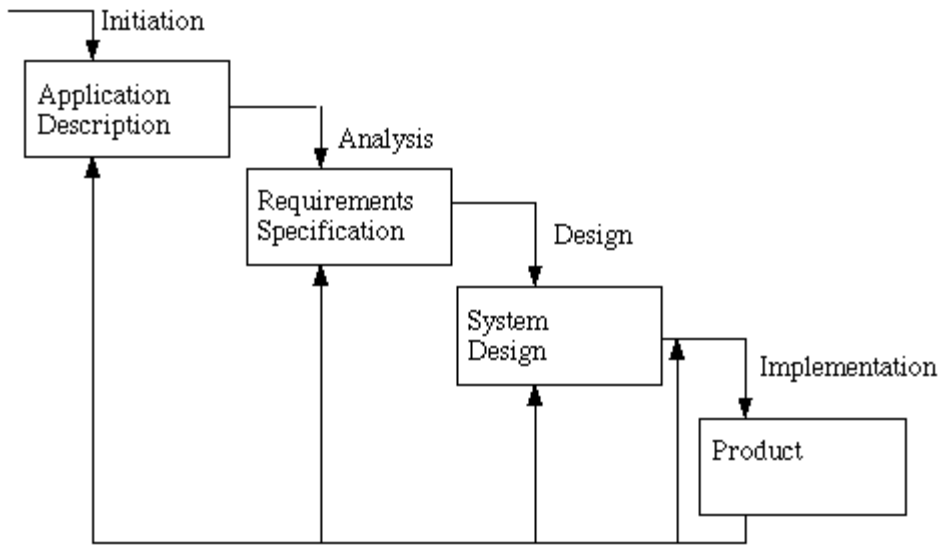


d. Quy trình phát triển

- Bao gồm thiết kế và kỹ thuật
- Các tiếp cận thiết kế
- Kỹ thuật và công cụ cài đặt
- Kỹ thuật đánh giá
- Hệ thống mẫu và Case studies



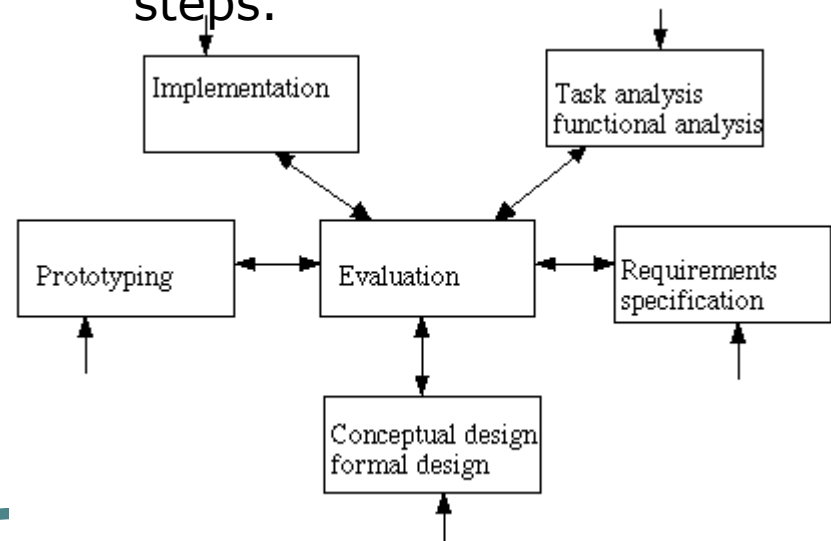
Quy trình phát triển



Model assumes that the design is fixed before entering to the next phase of design.

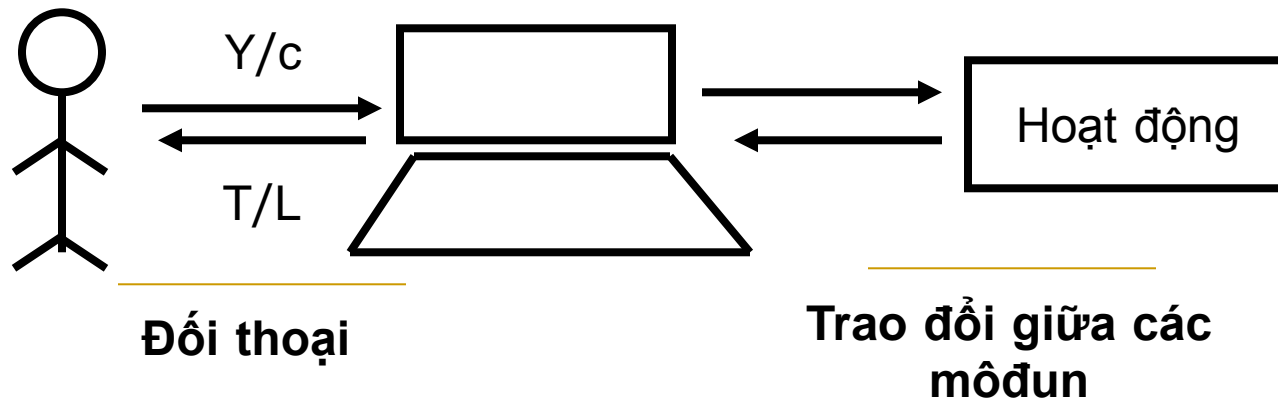
HCI Hix Hartson 1993

The design of interactive systems typically does not follow a specific order of steps.

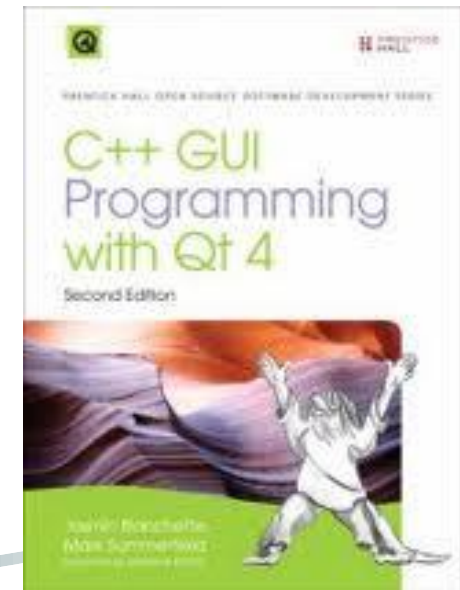
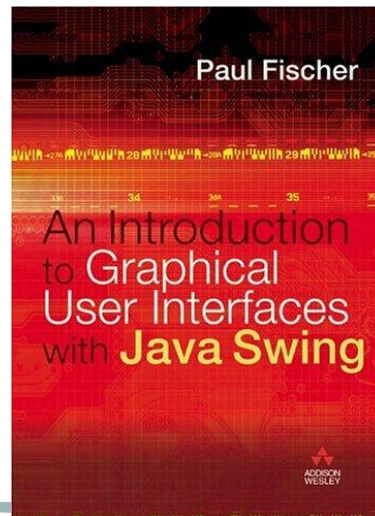
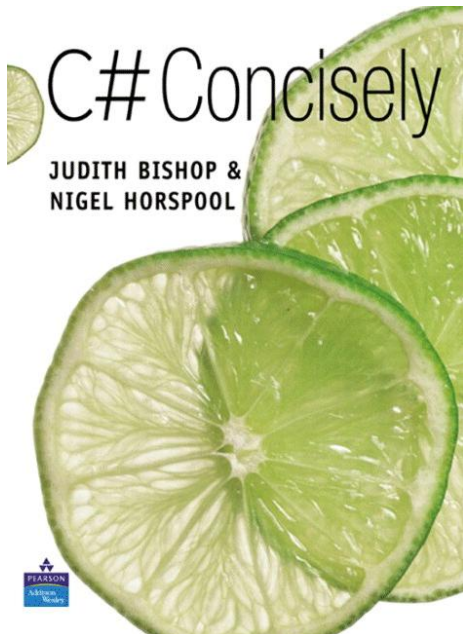
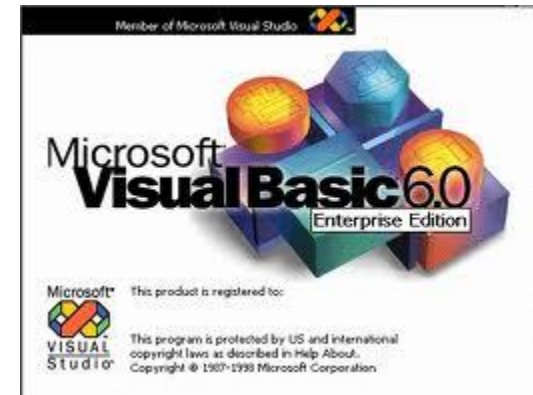
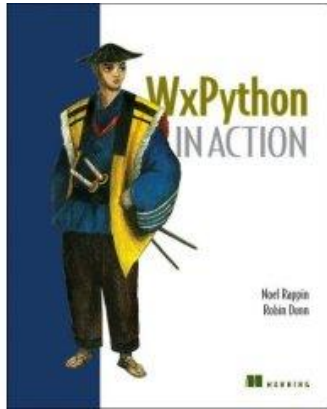


2. Phương tiện và công cụ tương tác

- Hai thành phần cơ bản: Con người và máy tính
- Giao tiếp:
 - Phương tiện: đối thoại thông qua môi trường (phần mềm)
 - Công cụ đối thoại: ngôn ngữ lập trình, thiết bị




Ngôn ngữ lập trình





3. Chất lượng tương tác

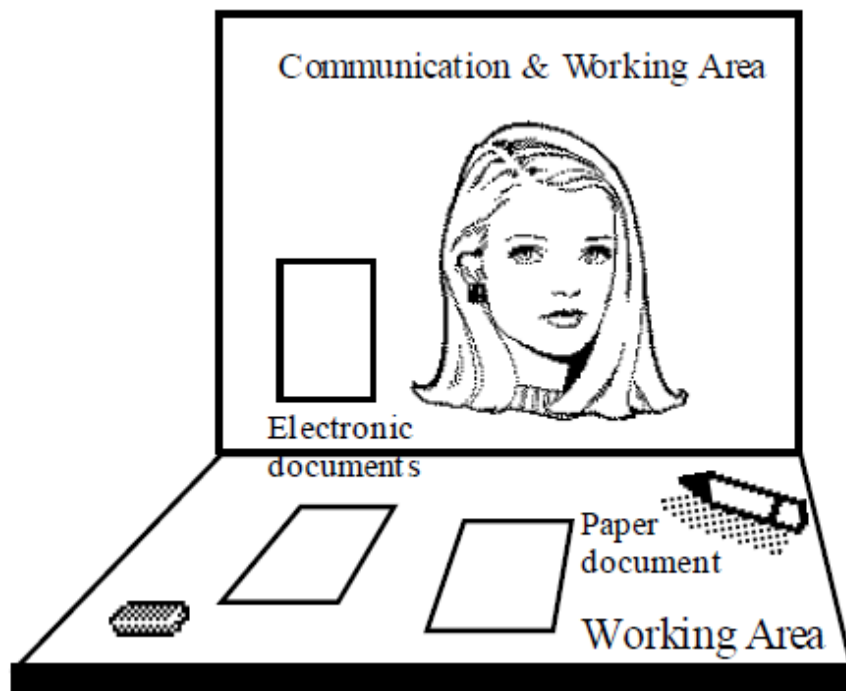
- Đầu tiên: Trạng thái ứng xử đúng với dữ liệu đúng
 - Tiếp theo: Trạng thái đúng với dữ liệu có thể sai
 - Hiện nay: Thân thiện, sinh động, dễ dùng
- 

4. Các lĩnh vực liên quan đến việc xây dựng các hệ tương tác

- Trước đây
 - Phương pháp tính
 - Mô hình toán học
 - Ngôn ngữ lập trình
- Hiện nay:
 - Phương pháp tính
 - Tính toán ký hiệu
 - Soạn thảo văn bản
 - Xử lý đồ họa, âm thanh, đa phương tiện
 - v.v.

IV. Ví dụ

- Giao diện tự nhiên



Hệ tương tác qua giao diện tự nhiên



Le Digital Desk.

À gauche, vue générale du prototype, à droite, détail de son utilisation.

Hệ tương tác qua giao diện tự nhiên: âm nhạc, giải trí, giám sát

The Reactable: a multi-touch interface for playing music. Performers can simultaneously interact with it by moving and rotating physical objects on its surface. Reactable was developed by Sergi Jordà and colleagues at the Universitat Pompeu Fabra, Barcelona. Icelandic songstress Björk used one on her 2007 tour. →



In Vodafone's vision of the future, young musicians will be able to create music with friends in remote places, all following or creating a musical score together. A wraparound screen shows video images of friends and displays the digital score. →



↑ Microsoft's 'Surface' is an interactive tabletop allowing two-handed interaction with digital objects such as photos, music files, games and maps. These kinds of interactive surfaces encourage collaborative, creative engagement.



↑ The Rovio robotic webcam is wirelessly connected to the Internet. It roams around the home providing an audio and video link to keep an eye on family or pets when you're out.

Source:
[Harmer10]

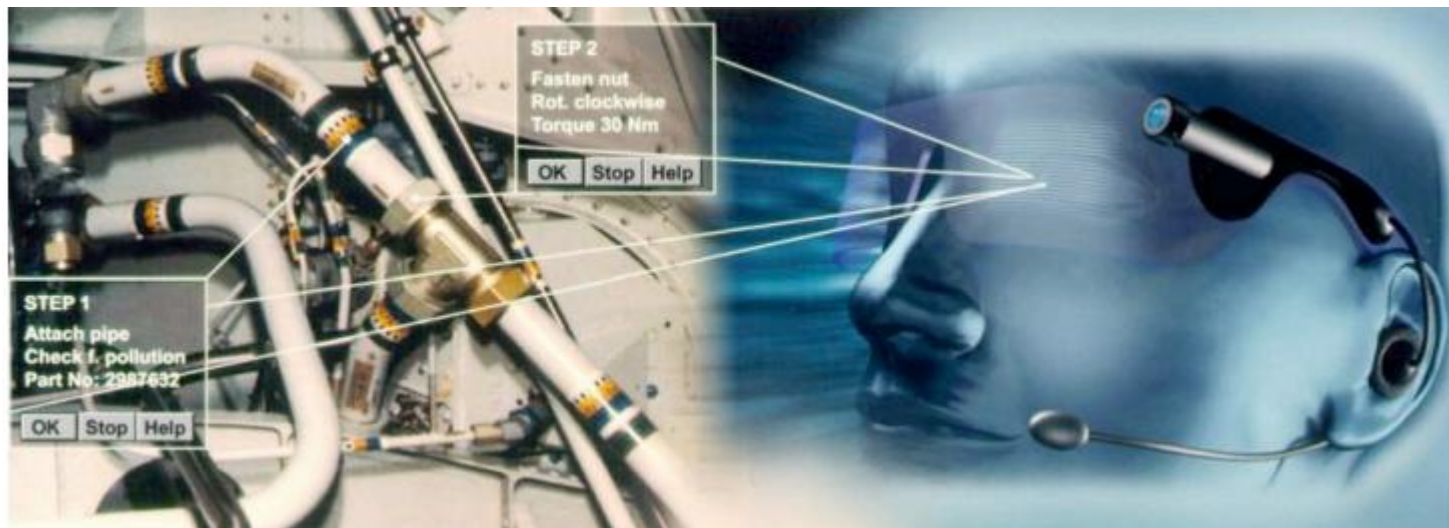
Hệ thống tính toán khắp nơi



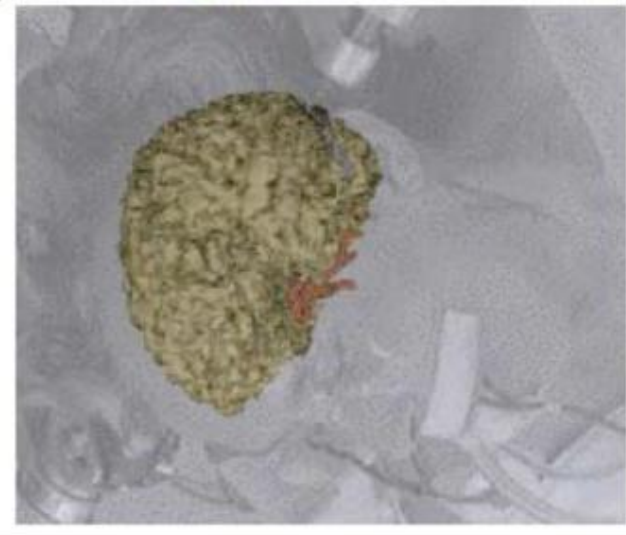
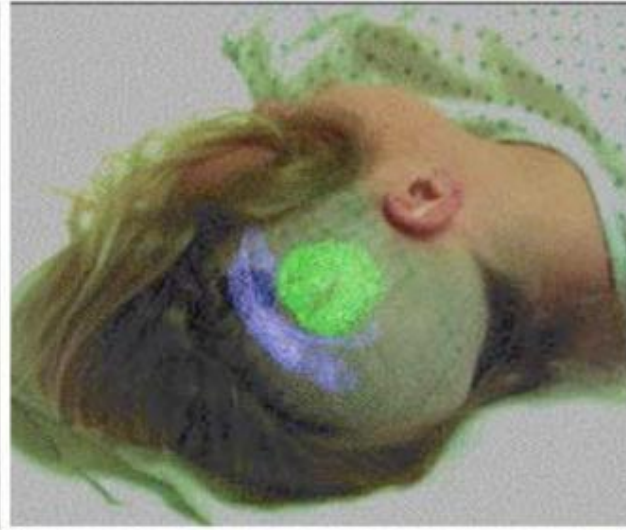
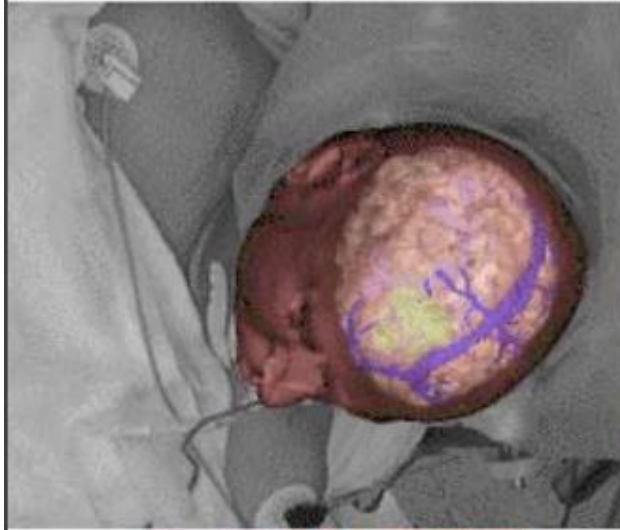
Phòng thông minh



Hiện thực ảo: game, lắp ráp, sửa chữa



Hiện thực ảo: ứng dụng trong y khoa






Questions



Tổng kết chương

- Một số định nghĩa, khái niệm cơ bản về tương tác người máy
 - Các yếu tố then chốt trong tương tác người máy
 - Lịch sử và các ứng dụng của tương tác người máy
- 



Bài tập về nhà



CHƯƠNG 2: CÁC YẾU TỐ THEN CHỐT TRONG TƯỞNG TÁC NGƯỜI MÁY

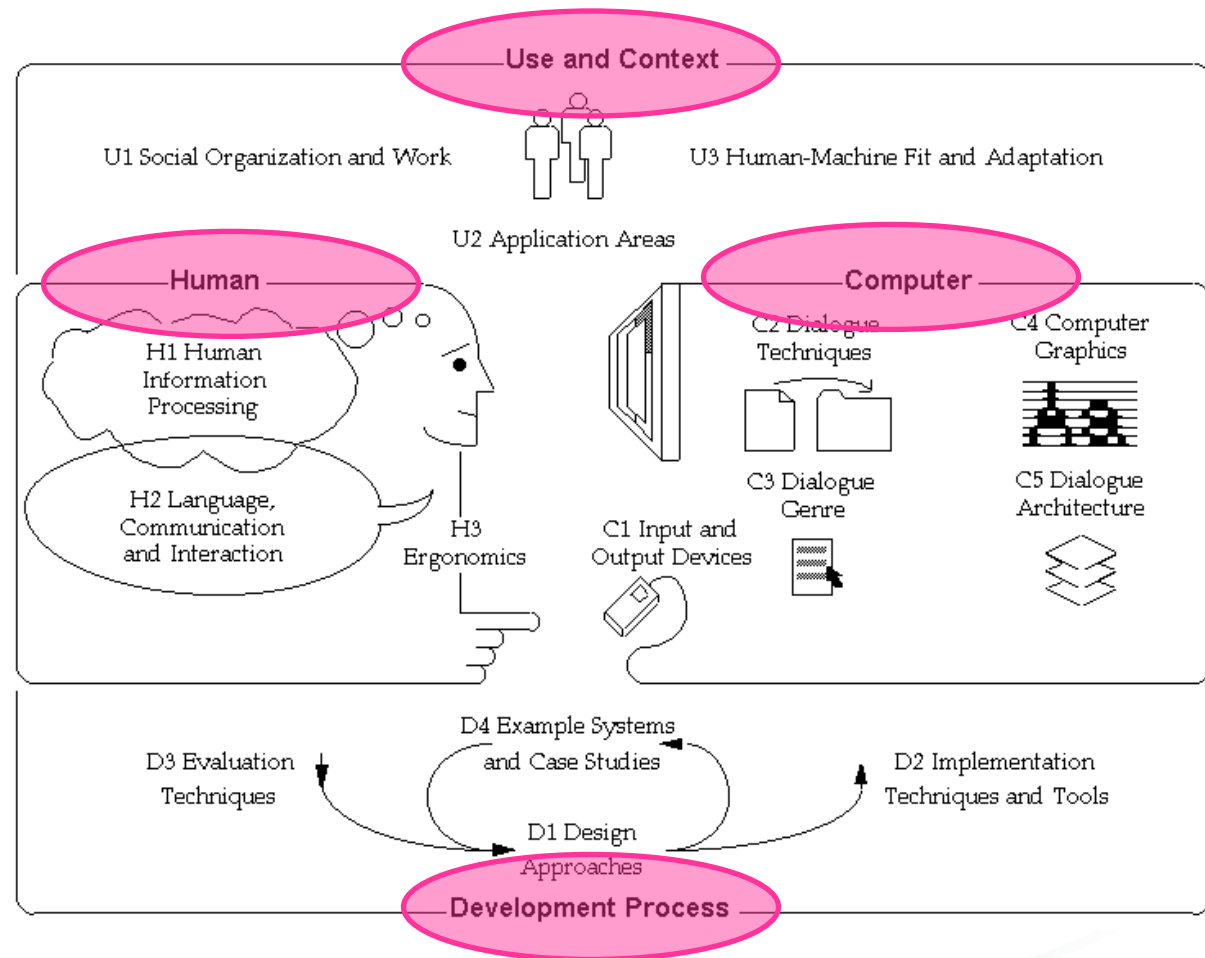
I. Con người

1. Tổng quan
2. Các kênh vào ra thông tin
3. Trí nhớ con người và ảnh hưởng đến quá trình giao tiếp
4. Lập luận và giải quyết vấn đề

II. Máy tính

Các thành phần chính của hệ tương tác

I. Con người 1. Tổng quan



Đặc điểm tâm sinh lý của con người

I. Con người

1. Tổng quan

- Con người có khả năng:
 - Thu nhận thông tin qua các hệ thống giác quan (Perceptual Systems)
 - Xử lý thông tin thông qua các hệ thống vận động (Motor systems)
 - Lưu trữ thông tin trong ký ức (Memory system)
- Cảm xúc, trạng thái ảnh hưởng đến khả năng của con người
- Khả năng của con người là hữu hạn
→ Ràng buộc khi thiết kế hệ tương tác cho người dùng con người

Mục đích nghiên cứu

- Tìm hiểu nhận thức và cách thức xử lý thông tin của con người
- Phân tích các khả năng của con người

Trên cơ sở đó, xây dựng các phần mềm tiện dụng

I. Con người

1. Tổng quan

2. Các kênh vào ra thông tin



visual, auditory



smell



taste



tactile



proprioception



haptics

2. Các kênh vào ra thông tin

Thị giác
Thính giác
Xúc giác
Vị giác
Khứu giác
Cảm nhận
Hành động căn
cứ vào xúc giác



Nói

Bằng lời

Không lời

Thơ

Biểu cảm trên khuôn mặt

Cử chỉ tay

Cử chỉ thân thể

Cử động mắt

Điều khiển hơi thở

Điều khiển thần kinh

EEG: Electroencephalography

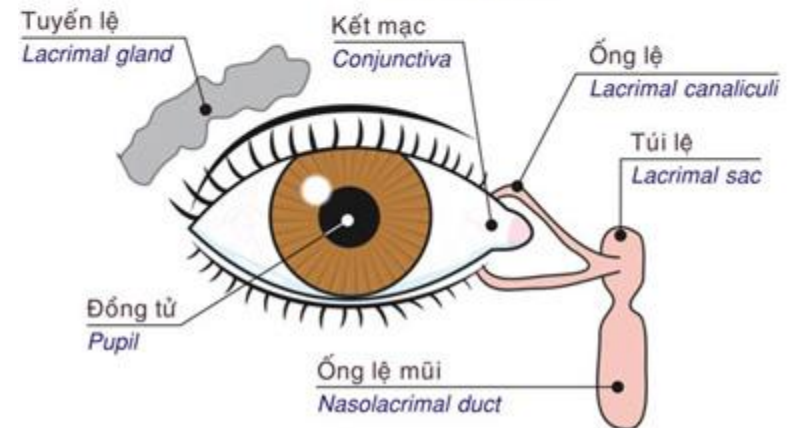
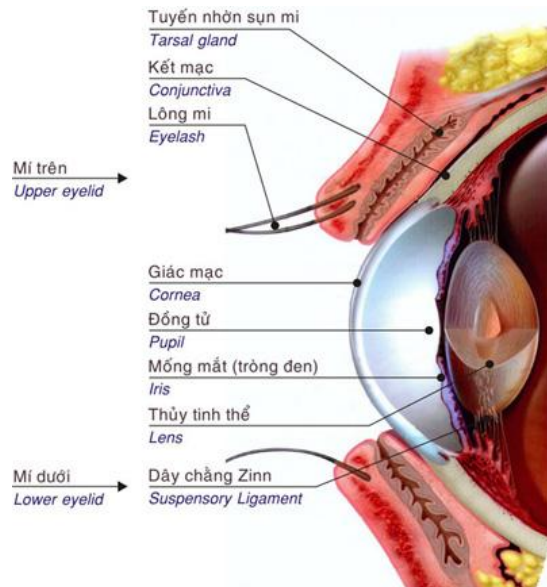
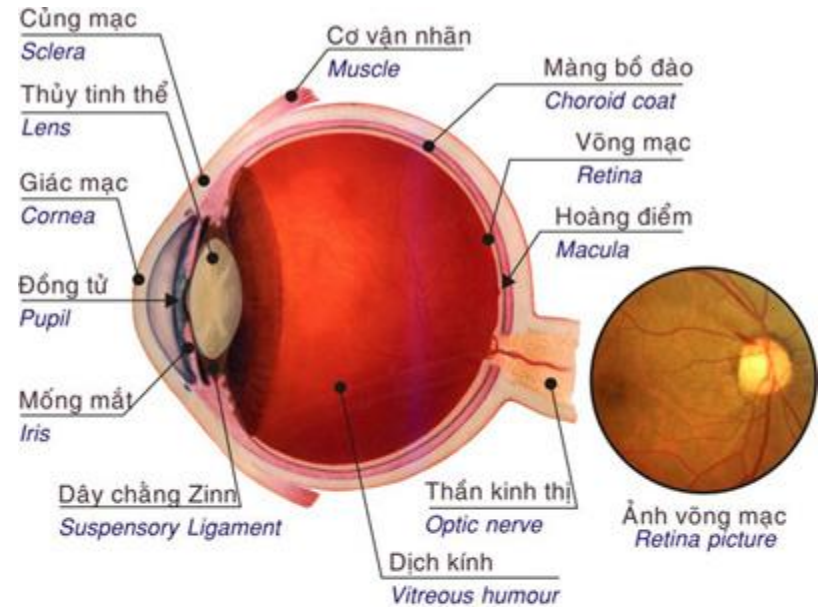
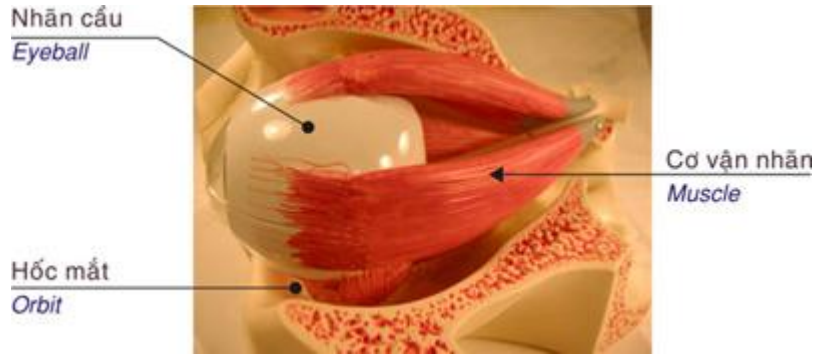
Tín hiệu sinh học

Nhịp tim

EMG: Electromyography

GSR: Galvanic Skin Response

2.1. Thị giác





Tương tác qua thị giác

- Xem xét sự phụ thuộc của cảm nhận thị giác vào
 - Kích thước hay khoảng cách tương đối giữa đối tượng quan sát và mắt
 - Độ sáng và độ tương phản của đối tượng
 - Khả năng và hạn chế của hệ thống thị giác

a. Cảm nhận ánh sáng

- Trên võng mạc có hai loại tế bào
 - Tế bào hình que: nhạy cảm với ánh sáng, cho phép nhìn thấy đối tượng trong điều kiện ánh sáng yếu
 - Tế bào hình nón: kém nhạy cảm với ánh sáng
 - Có ba loại tế bào hình nón cho phép cảm nhận ánh sáng với bước sóng khác nhau, giúp ta cảm nhận được màu sắc: đỏ, xanh lá cây và xanh lam

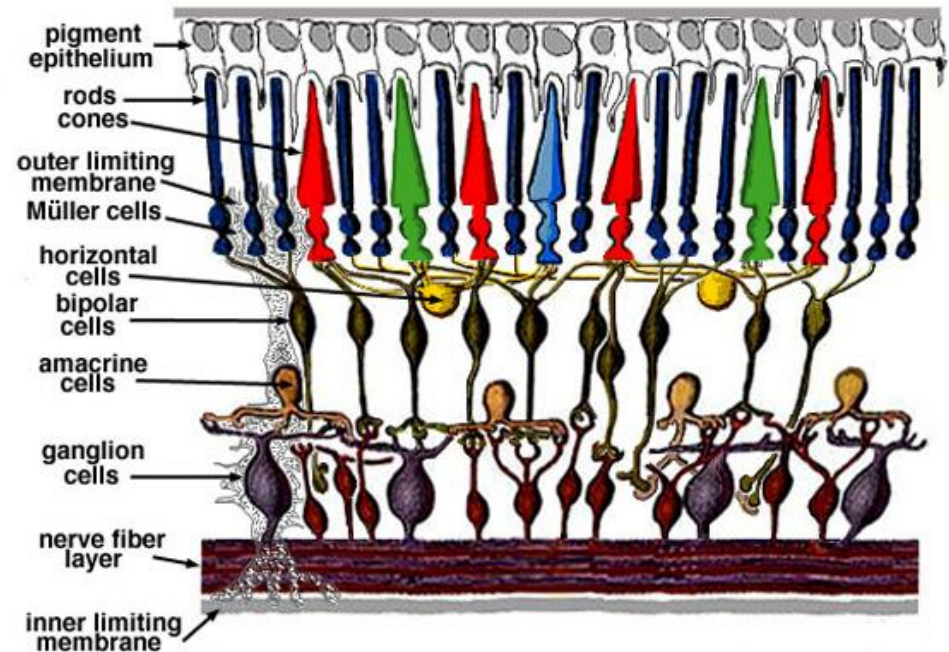
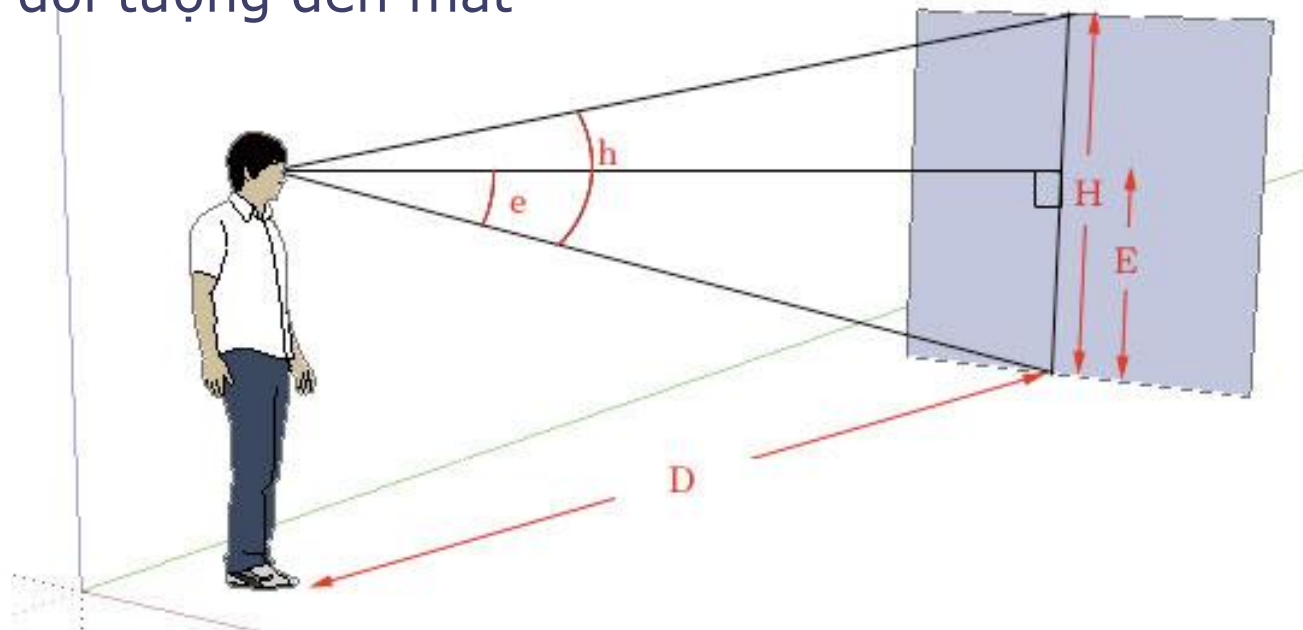


Fig. 2. Simple diagram of the organization of the retina.

b. Cảm nhận về kích thước

- Góc nhìn:

- là góc giới hạn bởi hai đường thẳng từ đỉnh và chân đối tượng đi qua tâm nhìn
- thường được đo bằng độ/phút/giây
- Phụ thuộc vào kích thước của đối tượng và khoảng cách từ đối tượng đến mắt



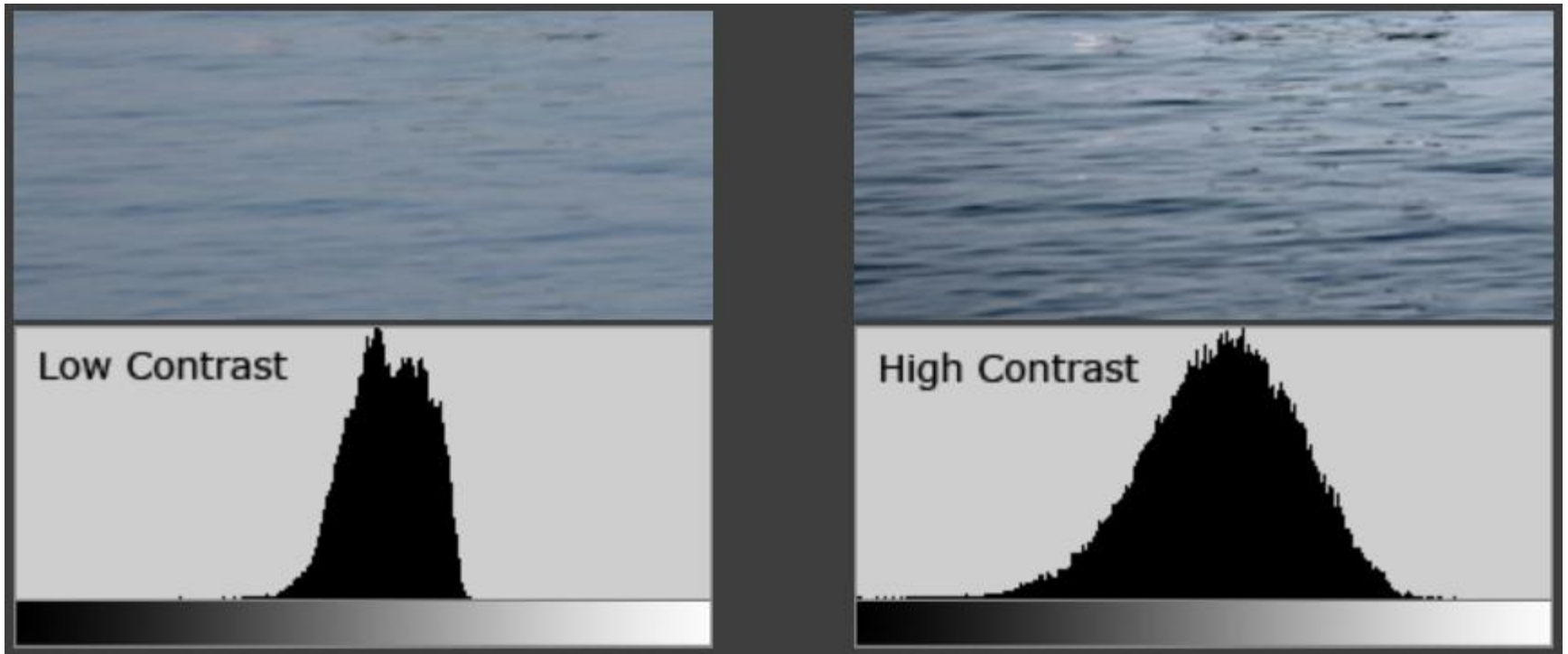
Cảm nhận về kích thước

- Hai đối tượng cùng khoảng cách, đối tượng nào có kích thước lớn thì sẽ nhìn tốt hơn (ảnh trên võng mạc sẽ lớn hơn)
 - Như vậy:
 - Nếu góc nhìn quá nhỏ: không có cảm nhận về đối tượng
 - Độ nhìn: Khả năng một người bình thường cảm nhận được các chi tiết của đối tượng
 - Luật hằng số của kích thước: ví dụ: cảm nhận về chiều cao của con người là không đổi, cho dù họ có chuyển động ra xa hoặc lại gần
- Sự cảm nhận về kích thước liên quan đến các yếu tố khác hơn là góc nhìn

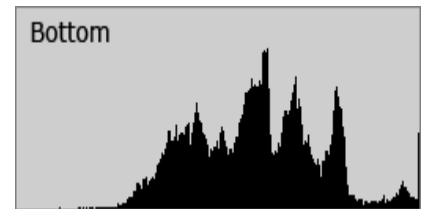
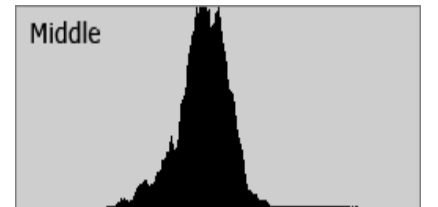
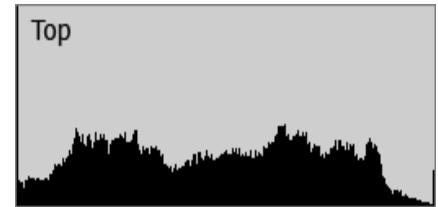
c. Cảm nhận độ sáng tối

- Độ sáng tối là đáp ứng chủ quan của mức độ sáng
 - Phụ thuộc vào:
 - Số tia sáng phát ra từ đối tượng
 - Tính chất phản xạ của bề mặt đối tượng
 - Độ tương phản: độ nổi của đối tượng so với nền
 - Hệ thống thị giác có khả năng tự điều chỉnh với các thay đổi về độ sáng tối
 - Độ nhìn tăng khi mức sáng tăng
 - Khi mức sáng tăng thì độ lập lờ cũng tăng
- Chú ý khi sử dụng các thiết bị hiển thị với mức sáng cao

Cảm nhận độ sáng tối

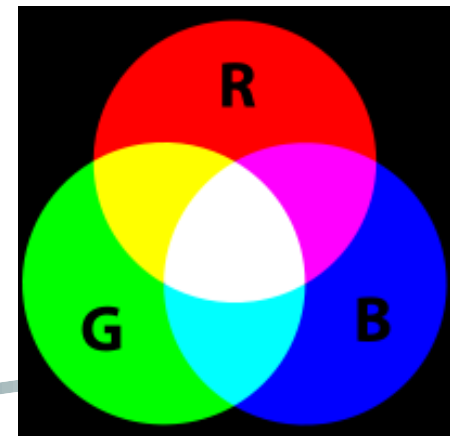
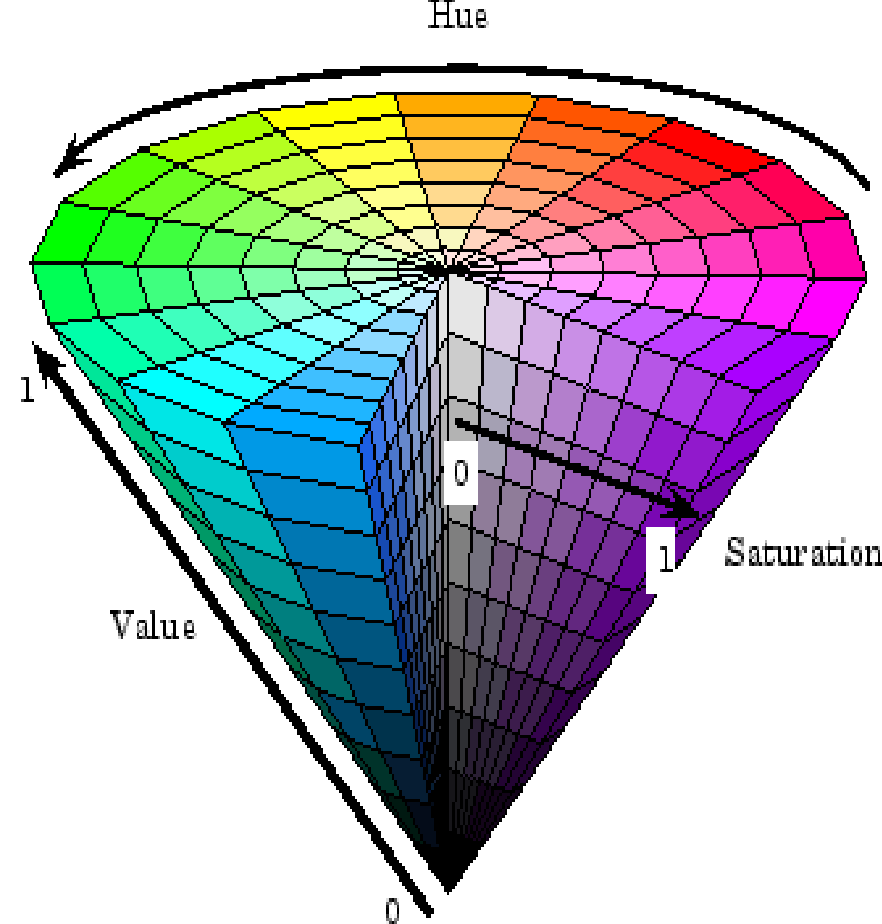


Cảm nhận độ sáng tối



d. Cảm nhận màu

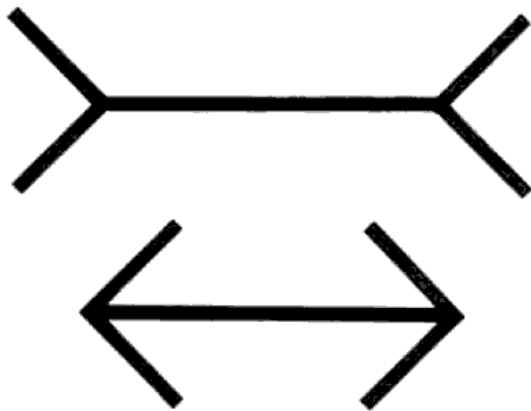
- 3 thành phần cơ bản
 - Hue: Sắc thái của màu
 - Intensity: Cường độ màu
 - Saturation: Độ bão hòa
- Nguồn sáng đơn sắc:
 - Sắc thái phụ thuộc vào bước sóng của ánh sáng
 - Độ bão hòa thay đổi nếu tăng lượng ánh sáng trắng
- Số màu mà mắt có thể cảm nhận được: hàng triệu màu
- Có một số trường hợp mù màu (8% đàn ông, và 1% phụ nữ)



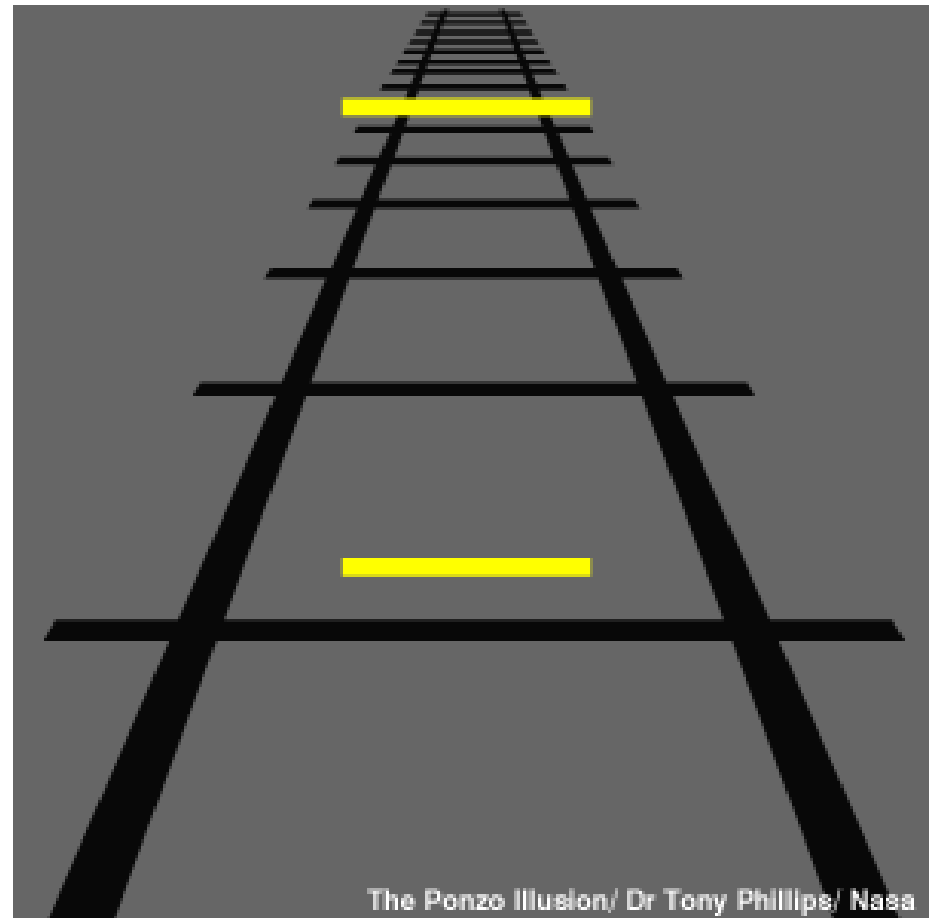
Khả năng, hạn chế của cảm nhận màu

- Phụ thuộc vào ý thức chủ quan
- Phụ thuộc vào chuyển động tương đối
- Hệ thống thị giác có khả năng điều chỉnh để ảnh hiện rõ trên võng mạc
- Màu sắc, độ sáng tối cũng thường được cảm nhận theo hằng số, bất chấp sự thay đổi của độ sáng
- Vì thế mắt người có thể giải quyết được một số trường hợp nhập nhằng
- Tuy nhiên sự điều chỉnh có thể dẫn đến ảo giác đánh lừa

Hiệu ứng Muller layer và hiệu ứng Ponzo



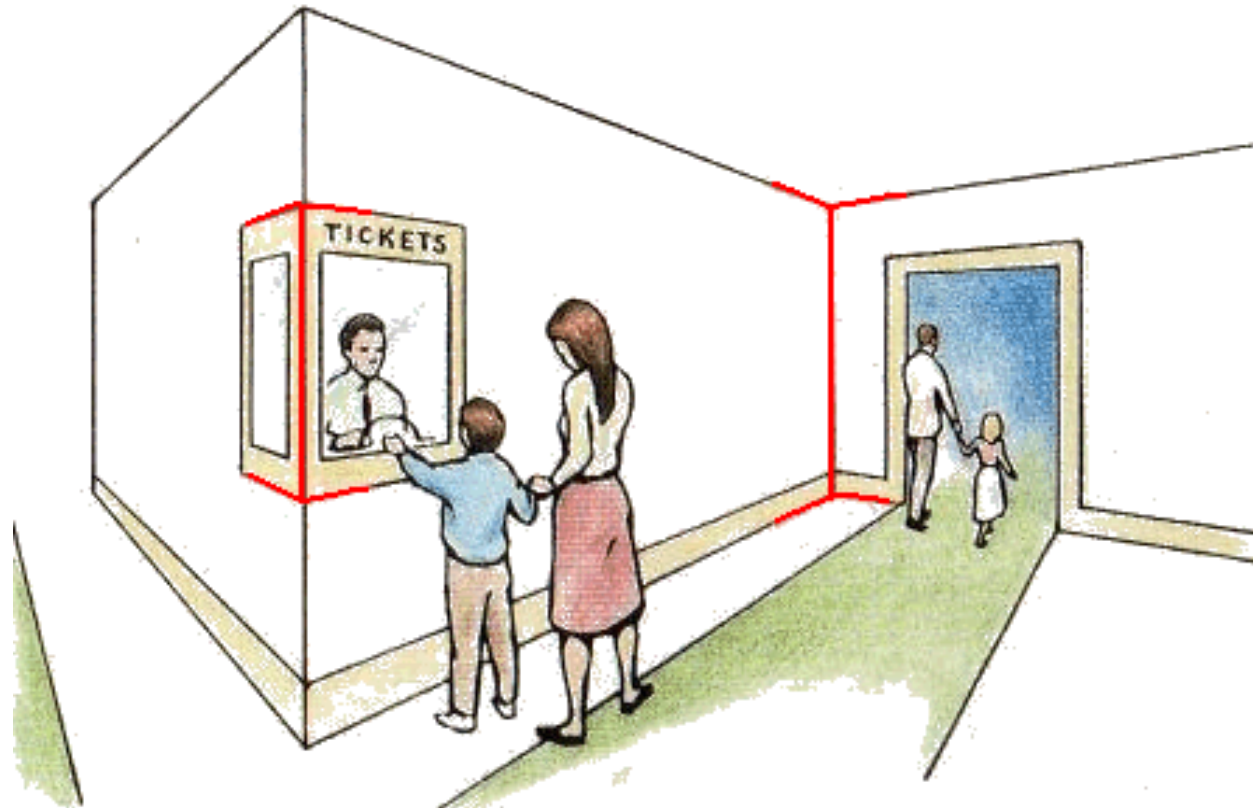
Muller layer: Đường nào dài hơn ?



The Ponzo Illusion/ Dr Tony Phillips/ Nasa

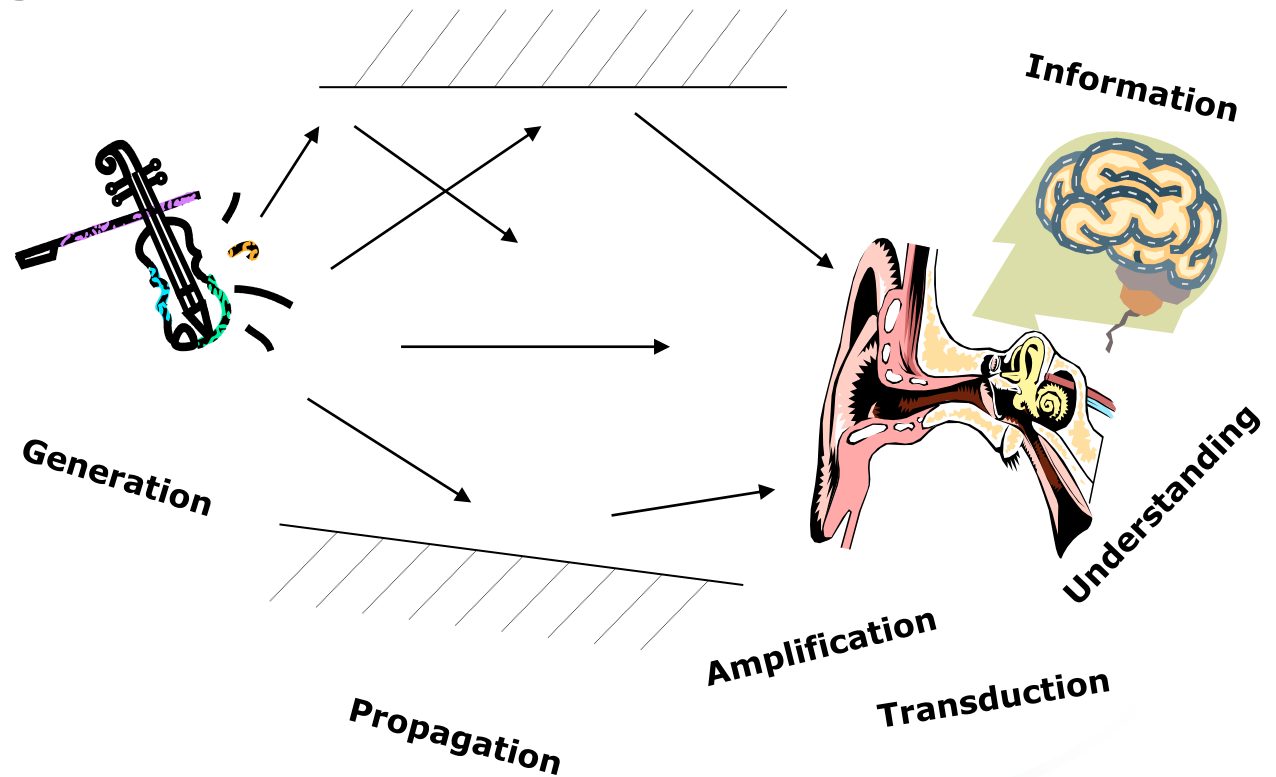
Ponzo: Đường nào dài hơn ? 17

Kết hợp hiệu ứng Ponzo và Muller-Layer



2.2. Thính giác

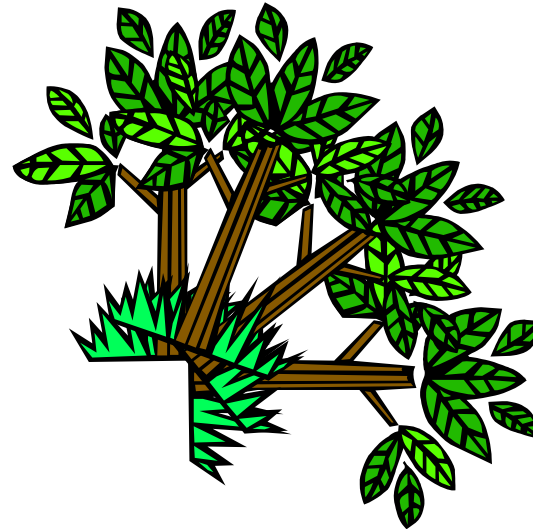
- Là giác quan thứ hai, song ít được sử dụng hơn thị giác trong HCI



Cảm nhận về âm thanh

- Cảm nhận âm thanh
 - Âm thanh là gì ?
 - Độ vang, tần số, âm sắc
 - Tai người có thể nghe trong khoảng : 20Hz – 15KHz
 - Các tần số khác nhau sẽ kích thích các hoạt động trên các vùng khác nhau của não và gây nên các xung động khác nhau.
 - Khi cảm nhận thông tin, hệ thống thính giác cần phải tiến hành lọc để loại bỏ tiếng ồn chỉ giữ lại thông tin hữu ích

If a tree falls in the forest and nobody is there to hear it, will it make a sound?





Thiết kế âm thanh trong tương tác

- Bài tập tại lớp: Đề xuất một số ý tưởng thiết kế giao diện tương tác thông qua âm thanh một cách hiệu quả
 - Speed sound
 - Non-speed sound



2.3. Xúc giác

- Xúc giác: giác quan thứ 3 trong tương tác người - máy
- Cung cấp
 - các thông tin có tính sống còn về môi trường
 - phương tiện chính trong hồi đáp
- Việc cảm nhận thông qua da:
 - Cảm nhận nhiệt
 - Cảm nhận sức căng do áp suất
 - Cảm nhận cơ khí: đáp ứng nhanh và đáp ứng chậm



Bài tập



2.4. Dịch chuyển

- Nhấn phím để đáp ứng một yêu cầu
 - Kích thích được cảm nhận bởi hệ thống cảm nhận
 - Truyền thông tin đến não
 - Xử lý yêu cầu
 - Khởi tạo đáp ứng
 - Não truyền lệnh đến các bộ phận tương ứng
- Mỗi hành động cần một thời gian: thời gian phản ứng và thời gian dịch chuyển
- Thời gian phản ứng phụ thuộc vào các thể : tình trạng sức khỏe, tuổi tác

Luật Fitt về sự dịch chuyển

- Tốc độ và độ chính xác của chuyển động là hai yếu tố quan trọng khi thiết kế hệ tương tác
- Ví dụ: Xem xét thời gian chi phí để dịch chuyển đến một đích cụ thể trên màn hình (phím lệnh, biểu tượng trên menu, etc ...)

- D : khoảng cách dịch chuyển

- L: kích thước đích

- X_i : k/c dịch chuyển sau i lần

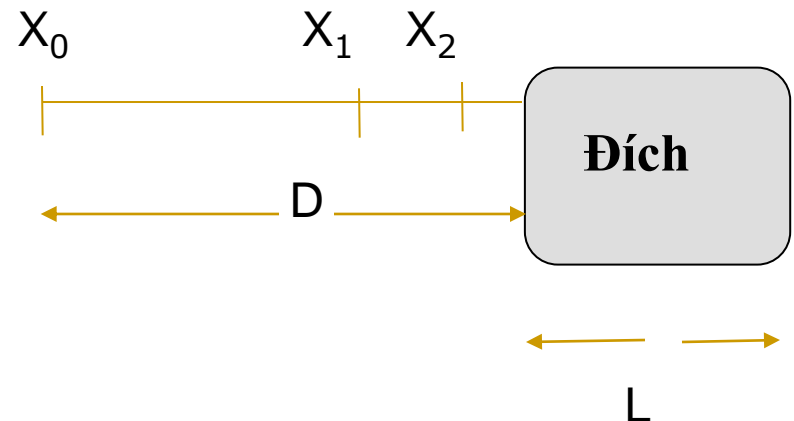
$$X_i = \varepsilon \times X_{i-1}$$

- Vì $X_0 = D \Rightarrow X_n = \varepsilon^n \times D$

- $X_n \leq L/2 \Rightarrow \varepsilon^n \times D \leq L/2$

- $n = -\log_2 (2D/L) / \log_2 \varepsilon$

$\tau = \tau_s + \tau_c + \tau_m$

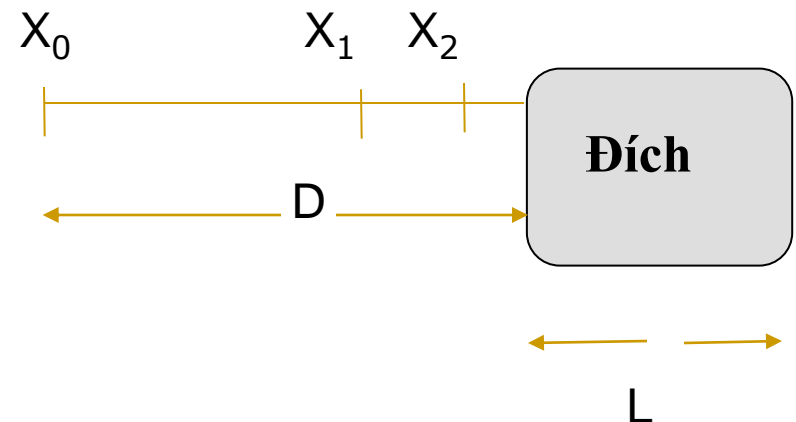


$$T = I \log_2 2D/L \text{ với :}$$
$$I = -(\tau_s + \tau_c + \tau_m) / \log_2 \varepsilon \approx 100 \text{ ms}$$

Luật Fitt về sự dịch chuyển

- Tốc độ và độ chính xác của chuyển động là hai yếu tố quan trọng khi thiết kế hệ tương tác
- Ví dụ: Xem xét thời gian chi phí để dịch chuyển đến một đích cụ thể trên màn hình (phím lệnh, biểu tượng trên menu, etc ...)




Thời gian chạm đích chỉ phụ thuộc vào tỉ số D/L (khoảng cách phải nhỏ, đích phải lớn)



$$T = I \log_2 2D/L \text{ với :}$$
$$I = -(\tau_s + \tau_c + \tau_m) / \log_2 \varepsilon \approx 100 \text{ ms}$$

Novint: <http://home.novint.com/index.php> Pioneering Interactive 3D touch product


Pioneering Interactive 3D Touch Products NVNT (OTC:BB): get quote | Stay In Touch

THE ORANGE BOX   


USERNAME:
PASSWORD: [Forgot Password?](#)

NOVINT PRODUCTS GAMES NEWS SUPPORT COMMUNITY STORE

Novint Falcon Pistol Grip Bundle THE ORANGE BOX






The Orange Box features three highly-awaited new games by Valve: Half-Life® 2: Episode Two, the second installment in the Half-Life 2 episodic trilogy;

100% MONEY BACK GUARANTEE on Falcon purchases if you're not completely satisfied 

[Click for more details](#)

NOVINT Games

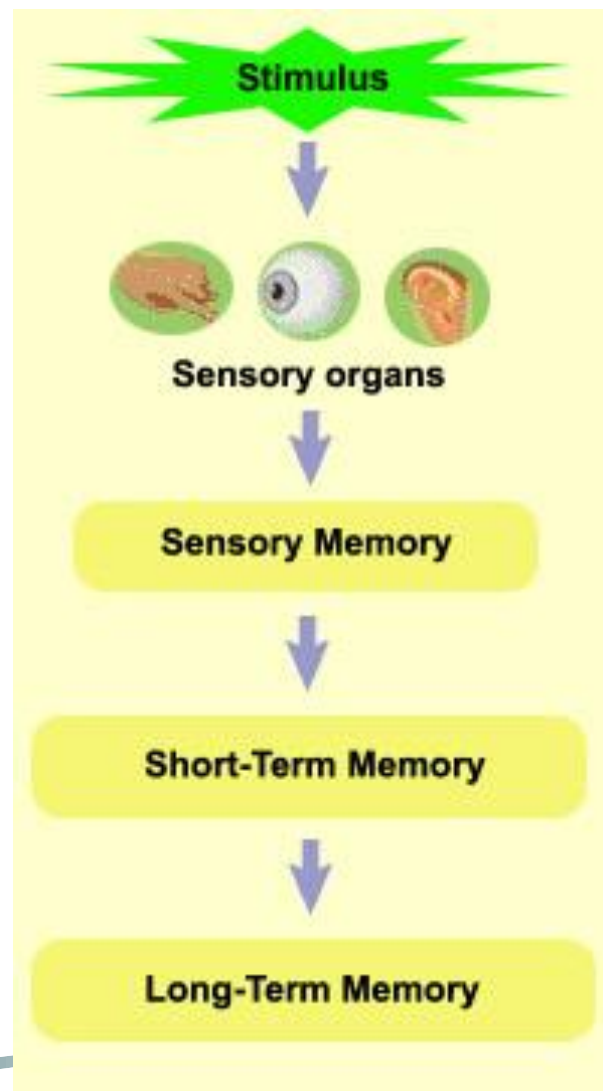
-  **The Orange Box**
5 games, one box.
-  **Tiger Woods PGA TOUR® 08**
Take on the world in Tiger Woods PGA TOUR® 08
-  **Day of Defeat: Source**
In Day of Defeat: Source, choose to fight for either the Axis or the Allies in this online, realism focused World War II

Các loại bộ nhớ

1. Con người
1. Tổng quan
2. Các kênh vào/ ra thông tin của con người

3. Trí nhớ con người và ảnh hưởng tới quá trình giao tiếp

- Bộ nhớ cảm nhận (sensory memory)
- Bộ nhớ ngắn hạn: hay bộ nhớ làm việc (short term memory)
- Bộ nhớ dài hạn: (longterm memory)





Ví dụ

- Trong vòng 3 giây, hãy thử tìm xem có bao nhiêu số 3 trong văn bản sau:

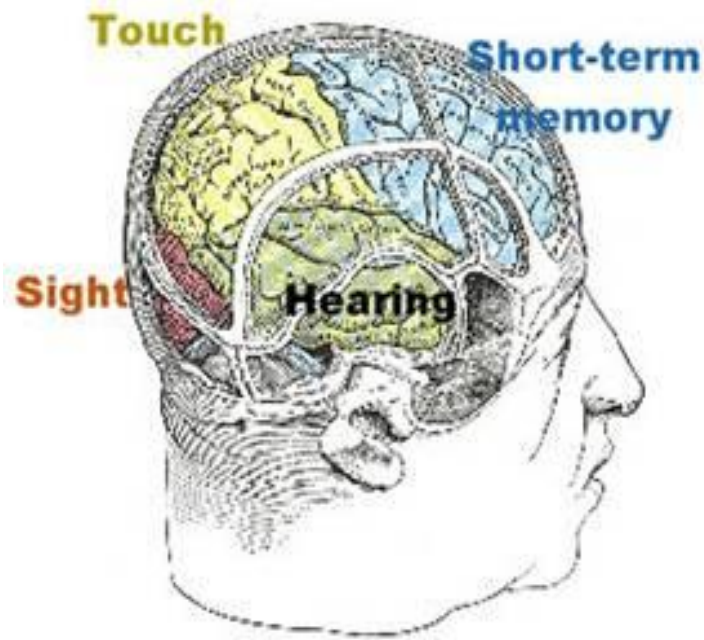


3.1. Bộ nhớ cảm nhận

- Bộ nhớ cảm nhận: chứa các kích thích nhận được từ các giác quan như nghe, nhìn, sờ mó
- Mỗi giác quan có bộ nhớ cảm nhận riêng tại đó các kích thích được mã hóa
- Thông tin trong bộ nhớ cảm nhận được lưu theo cách viết đè
- Thông tin từ bộ nhớ cảm nhận được đưa qua bộ lọc để chuyển đến bộ nhớ ngắn hạn
- Thời gian lưu lại thông tin
 - Bộ nhớ thị giác: 200ms
 - Bộ nhớ thính giác: 1500ms

3.2. Bộ nhớ ngắn hạn

- Thời gian truy cập: nhanh (700ms)
- Hư hỏng: nhanh (200ms)
- Khả năng hạn chế: 7 ± 2 chữ số

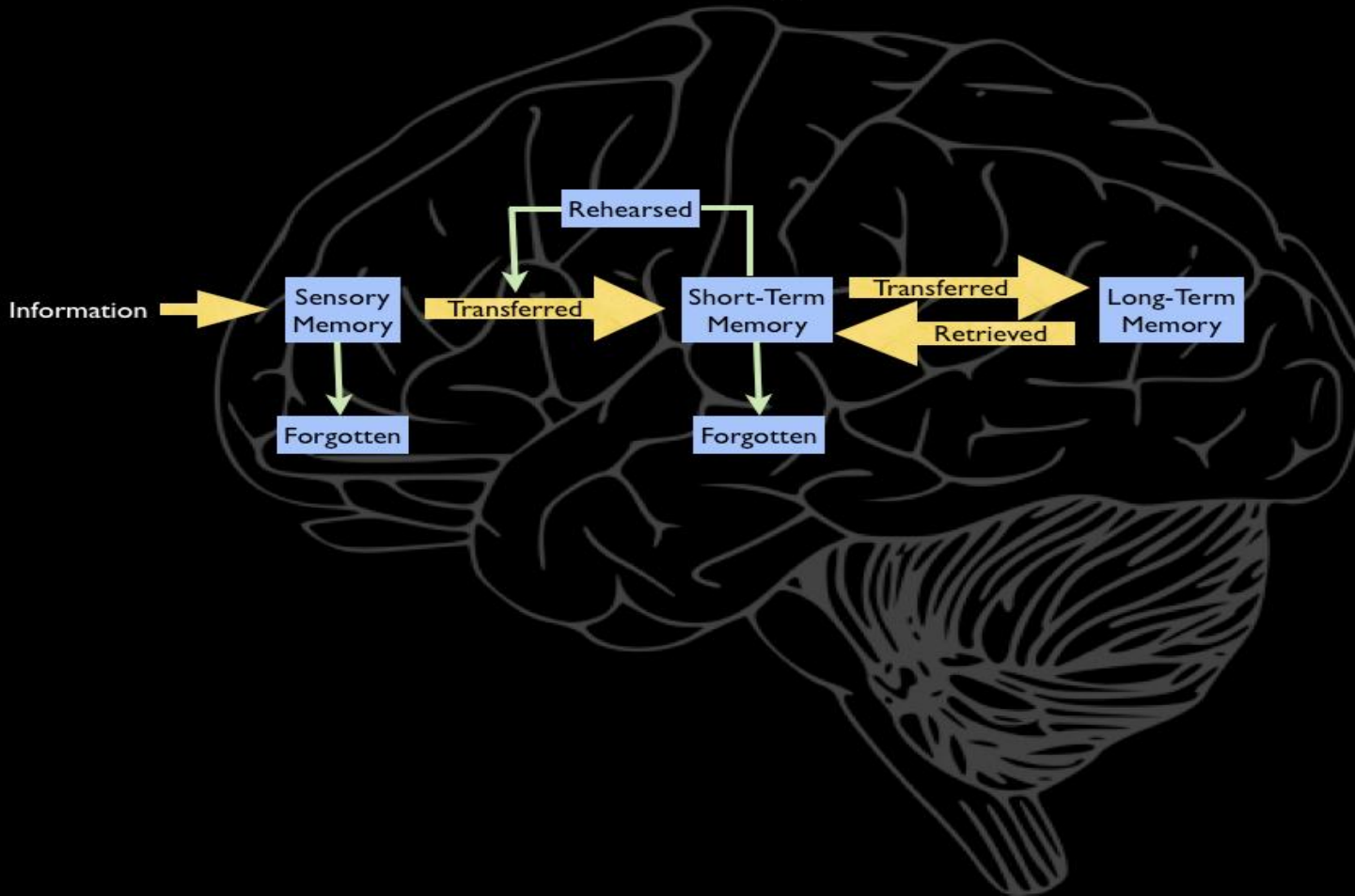


3.3. Bộ nhớ dài hạn

- Các đặc trưng của bộ nhớ dài hạn:
 - Cấu trúc tuyến tính
 - Truy nhập chậm: 1/10 s
 - Hư hỏng: chậm
 - Khả năng không hạn chế
 - Thông tin biểu diễn qua mạng ngữ nghĩa, frame, hoặc luật sản xuất, . . .
- Hai kiểu bộ nhớ LTM:
 - Rời rạc (Episodic): bộ nhớ tuần tự các sự kiện
 - Ngữ nghĩa (Semantic): bộ nhớ có cấu trúc của các sự kiện, khái niệm và kiểu.
- Thông tin trong bộ nhớ ngữ nghĩa lấy từ bộ nhớ rời rạc

Quá trình xử lý thông tin

The Information Processing Model



4.1. Lập luận

- I. Con người
- 1. Tổng quan
- 2. Các kênh vào ra
- 3. Bộ nhớ
- 4. Lập luận và giải quyết vấn đề**

- Lập luận: Là một quá trình sử dụng tri thức đã có để dựng nên kết luận hay suy diễn điều mới trong lĩnh vực quan tâm
- Các kiểu lập luận
 - Suy luận
 - Quy nạp
 - Phản chứng
- Lấy ví dụ một số kiểu lập luận ?



Lập luận suy luận

- Dạng suy luận: IF [conditions] THEN [actions]
- Đưa ra kết luận cần thiết 1 cách logic từ các giả thiết:
 - If it is Friday then she will go to work
 - It is Friday therefore she will go to work.
- Kết luận logic không nhất thiết phải đúng
 - If it is raining then the ground is dry
 - It is raining therefore the ground is dry



Wason card problem

- Có 4 thẻ bài: A, B, 4, 7. Trên mỗi mặt của thẻ bài là một chữ cái và một chữ số ở mặt còn lại
- Hỏi cần phải lật mặt thẻ nào để xác định khẳng định sau là sai: Nếu một card có một nguyên âm ở một mặt, thì mặt sau của nó sẽ là một số chẵn





a. Suy luận quy nạp

- Quy nạp: suy diễn từ cái đã biết sang cái chưa biết
- Ví dụ: Nếu nhìn thấy một con voi có vòi => tất cả các con voi đều có vòi
- Cách suy luận này không phải lúc nào cũng đáng tin cậy
- Tuy nhiên con người thích suy luận tích cực hơn tiêu cực hoặc phủ định

b. Suy luận phản chứng

- Phản chứng: Đưa ra các giải thích về sự kiện quan sát
- Ví dụ: Sam thường lái xe rất nhanh khi say rượu
 - Khi gặp Sam lái xe nhanh \Rightarrow Sam đang say rượu
 - Điều này không phải lúc nào cũng đúng: Sam có thể lái xe nhanh do vội
 - Con người luôn suy luận như vậy: Nếu như có một sự kiện E xảy ra tiếp theo một hành động A \Rightarrow hành động A luôn gây ra sự kiện E.
 - \Rightarrow lưu ý trong thiết kế tương tác



4.2. Giải quyết vấn đề

- Lập luận: phương tiện để suy diễn thông tin mới từ cái đã biết
- Giải quyết: Quá trình tìm lời giải cho một nhiệm vụ chưa biết với các tri thức đã có
- Để giải quyết vấn đề: con người phải có khả năng thích nghi thông tin đã có để xử lý thông tin mới
- Cách thức giải quyết vấn đề
 - Lý thuyết Gestalt: ~1920
 - Lý thuyết không gian: ~1970

a. Lý thuyết Gestalt

1. Tính tương tự (similarity)
2. Tính gần nhau (proximity)
3. Tính liên tục (continuation)
4. Tính đóng (closure)



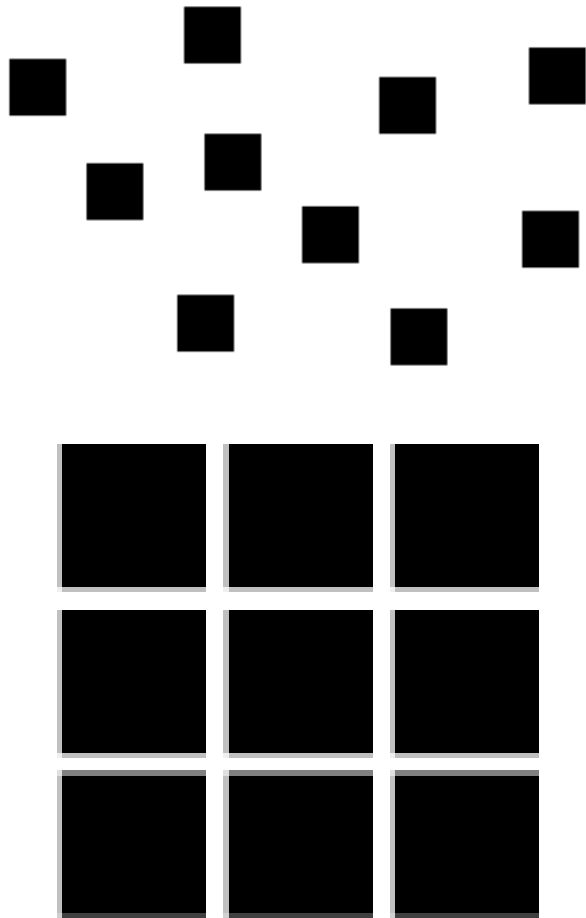
Christian Freiherr von Ehrenfels
1859 - 1932



Similarity



Proximity





Continuation

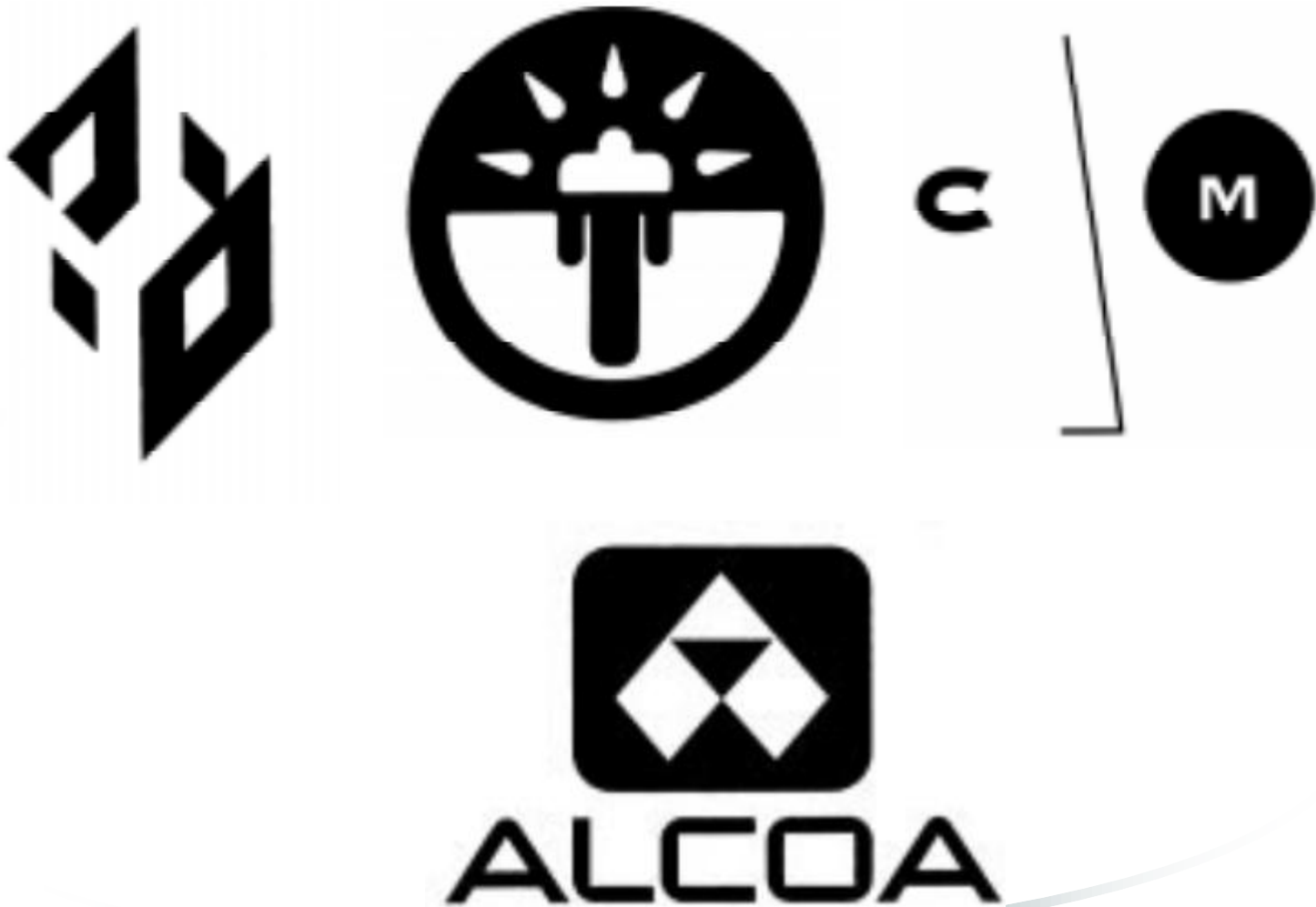




Closure



Hình sau sử dụng các nguyên lý Gestalt nào ?



Ứng dụng nguyên lý Gestalt trong thiết kế Web

CALIFORNIA OAK MORTALITY TASK FORCE

Search

Monthly Report:
[February 2004](#)

[COMTF Meeting on March 9-10, 2004](#)
[Information and Registration](#)
Details of the Spring COMTF meeting are now available. Click the link above for complete information and registration instructions.

[Upcoming Events](#)
Upcoming events include the Spring COMTF Meeting in March, and additional Recognition and Treatment Training courses in April and May.

[Sudden Oak Death Diagnostic Guide](#)

Sudden Oak Death and the California Oak Mortality Task Force

The Task Force focuses on the potentially devastating effects of a newly discovered pathogen called *Phytophthora ramorum*. It has caused an outbreak of Sudden Oak Death, which is killing large numbers of the native oaks and tanoaks in California's coastal regions.

Find information on:

- [Which counties are infested?](#)
- [What are the symptoms?](#)
- [What plants are affected?](#)
- [How do I report an infested site?](#)
- [What are the current regulations and quarantines?](#)
- [What is the most recent research?](#)
- [What is the Task Force?](#)
- [How do I sign up to receive monthly newsletters?](#)

Also...

Where is Sudden Oak Death in California? Is it in my county? Find out at the Sudden Oak Death Monitoring Site: [Where is SOD?](#)

How can I tell if my tree has Sudden Oak Death? Download the latest [SOD Diagnostic Guide](#) to find out.

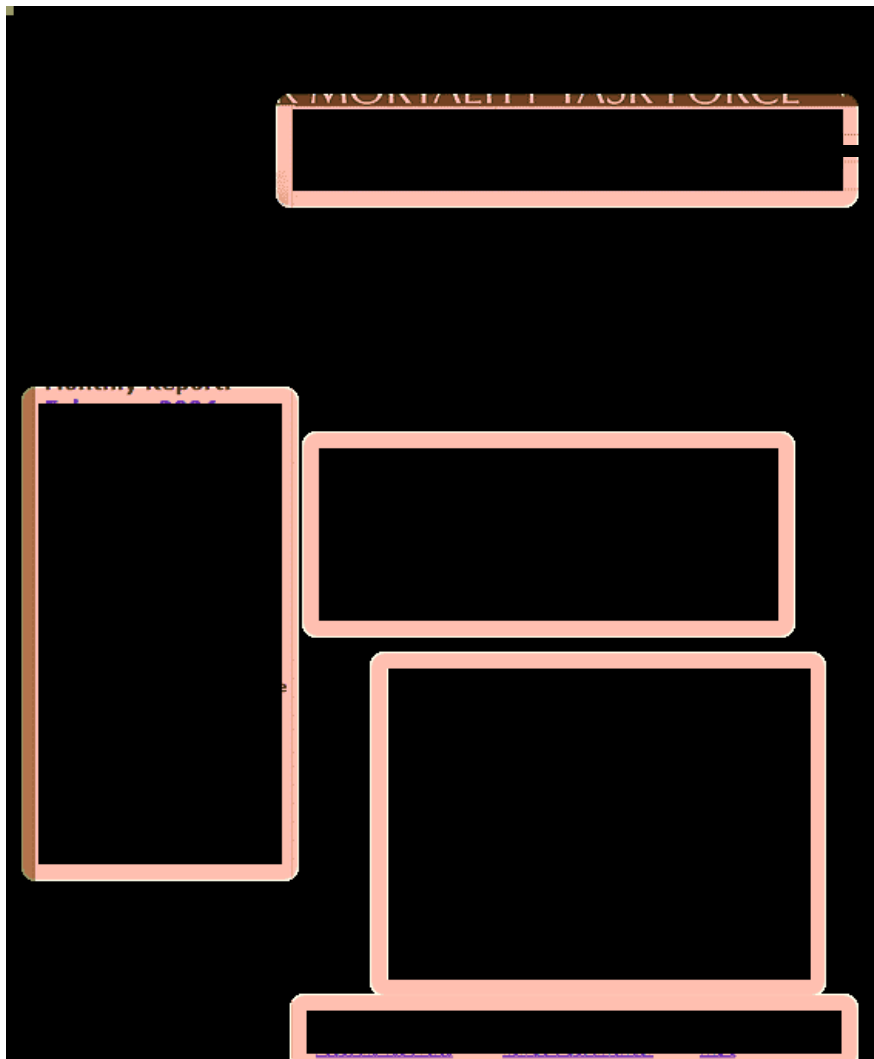
Add your oak to the SOD database, look at the local distribution, create a map -- all at the [OakMapper](#) web site.

Contact SODBusters at 1-866-SOD-7411, or check out their site at <http://groups.ucanr.org/SODBusters/>.
And much more.

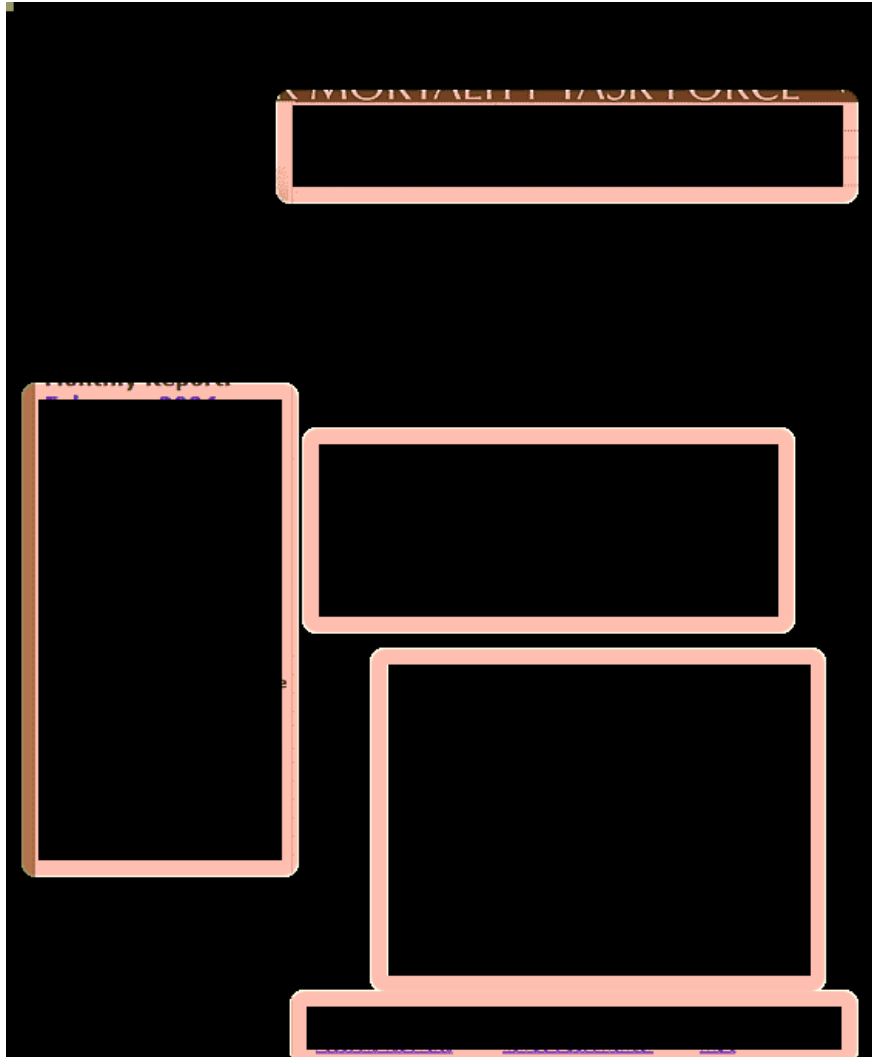
[About Sudden Oak Death](#) [For the Professional](#) [For the Homeowner](#)
[Publications & Resources](#) [Regulations](#) [Contacts](#)
[About the Task Force](#) [How Do I Get Involved?](#) [FAQ's](#)

Các nguyên lý nào được sử dụng trong việc thiết kế trang web này?

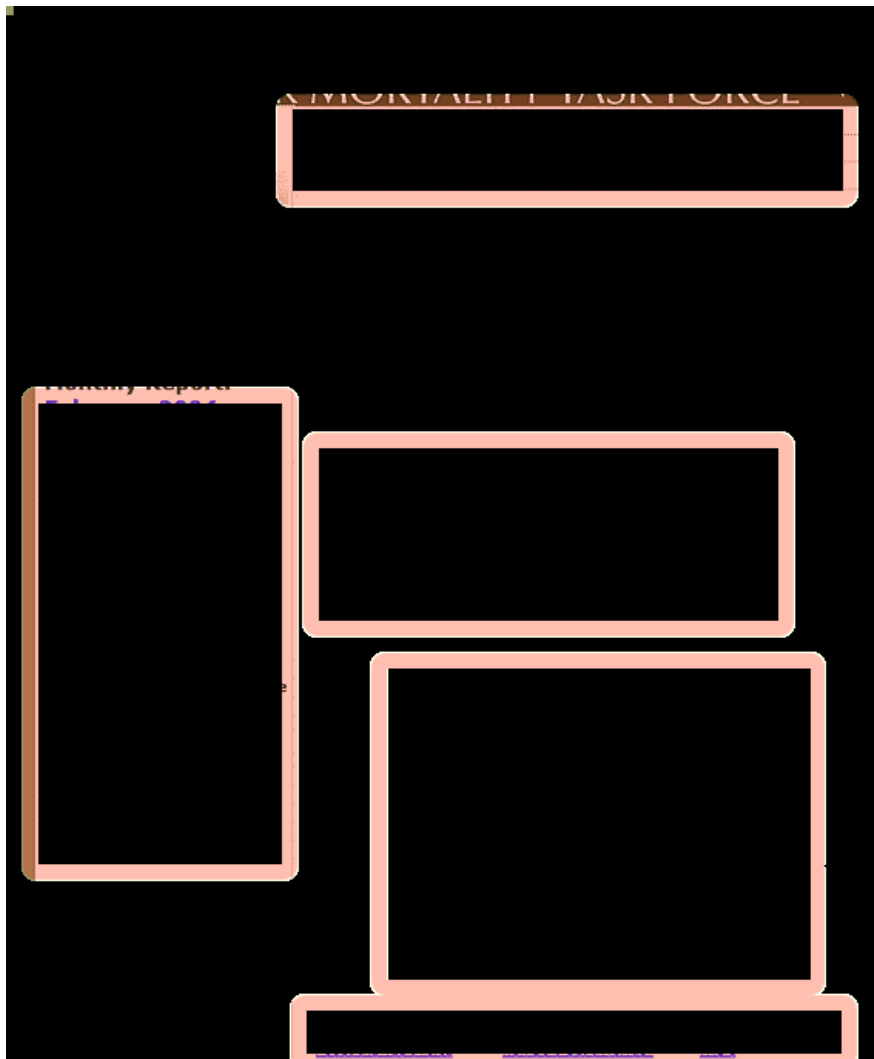
Ứng dụng nguyên lý Gestalt trong thiết kế Web



Ứng dụng nguyên lý Gestalt trong thiết kế Web



Ứng dụng nguyên lý Gestalt trong thiết kế Web



Ứng dụng nguyên lý Gestalt trong thiết kế Web

- Cung cấp các thông tin ít xuất hiện để giúp người đọc hình dung cách trang web tổ chức việc hiển thị nội dung trước khi người đọc tập trung vào các thông tin có tần suất xuất hiện cao trên trang web như văn bản và hình ảnh
- Điều này được minh chứng thông qua việc làm mờ nội dung chi tiết của trang web

Ứng dụng nguyên lý Gestalt trong thiết kế Web



Dễ dàng phân biệt các nhóm phần tử khác nhau mà không cần xem nội dung chi tiết

Ứng dụng nguyên lý Gestalt trong thiết kế Web



b. Lý thuyết không gian bài toán

- Newell and Simon: General Problem Solver (GPS)
- Tập trung vào không gian bài toán
 - Các phát biểu của bài toán
 - Giải quyết vấn đề: Khởi tạo các phép phát biểu này bằng các phép dịch chuyển hợp lý
 - Bài toán có trạng thái đầu, trạng thái đích
 - Để đạt đến trạng thái đích => cần thực hiện phép dịch chuyển
 - Sử dụng các heuristic để lựa chọn phép dịch chuyển

Chess: of Human and artificial intelligent



Deepblue



Garry Kimovich Kasparov

[Carnegie Mellon and move to IBM Research](#)



CHƯƠNG 2: CÁC YẾU TỐ THEN CHỐT TRONG TƯƠNG TÁC NGƯỜI MÁY

I. Con người

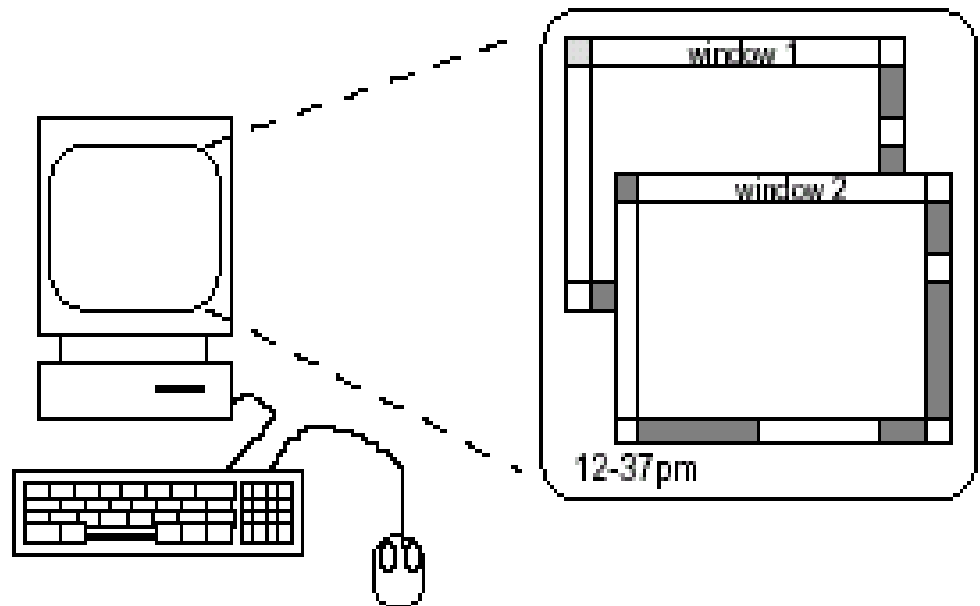
II. Máy tính

1. Máy tính với khả năng tương tác trực tiếp
2. Thiết bị vào
3. Thiết bị ra
4. Khả năng xử lý thông tin và ảnh hưởng tới quá trình tương tác

1. Tổng quan

- « Máy tính là một thành phần tham gia tương tác, thực hiện chương trình » - W. Thimbleby
- Các thành phần của một hệ thống máy tính tiêu biểu
 - Thiết bị vào ra
 - Bàn phím
 - Chuột
 - Màn hình
 - Loa
 - Bộ nhớ
 - RAM, HD, CD
 - Bộ xử lý
 - Batch
 - Online

Quan tâm đến ảnh hưởng của chúng tới thiết kế tương tác



a. Bàn phím

- QWERTY



- I. Con người
- II. Máy tính
 1. Tổng quan
 2. **Thiết bị vào**

a. Bàn phím

- Bàn phím chữ cái



- I. Con người
- II. Máy tính
- 1. Tổng quan
- 2. Thiết bị vào**

a. Bàn phím

- DVORAK



The Dvorak typewriter keyboard layout that was publicly promulgated



- I. Con người
- II. Máy tính
- 1. Tổng quan
- 2. **Thiết bị vào**

a. Bàn phím

- MALTRON: Thiết kế đặc biệt, tránh tổn thương cổ tay của người dùng



I. Con người
II. Máy tính
1. Tổng quan
2. Thiết bị vào

Bài tập

- Tìm điểm khác biệt trong nguyên tắc thiết kế:
 - Cho việc bấm phím nhanh
 - Thuận lợi cho việc nhớ (PIN Code)



Calculator



ATM keyboard



Mobile phone keyboard⁶¹

b. Chữ viết tay

- Nhận dạng chữ viết tay
- Đầu vào: chuyển thành file văn bản
- Khó khăn:
 - độ chính xác nhận dạng chưa cao
 - Số lượng cá thể lớn
- Ưu điểm:
 - Hệ thống dựa vào chữ viết thường nhỏ hơn
 - Các phím nhỏ thì khó dùng và chóng mệt



c. Tiếng nói

- I. Con người
- II. Máy tính
 - 1. Tổng quan
 - 2. Thiết bị vào**

- Nhận dạng tiếng nói
- Tốc độ nhận dạng: 90%
- Ưu điểm:
 - Tự nhiên
 - Dễ sử dụng
- Nhược điểm:
 - Thiếu chính xác
 - Sai lỗi chính tả

d. Thiết bị trở và định vị

- Chuột
- Trackball
- Joystick

I. Con người
II. Máy tính
1. Tổng quan
**2. Thiết bị
vào**



3. Thiết bị ra

- Màn hình:
 - CRT(Cathode Ray Tube): Dòng điện tử bắn ra từ súng điện tử và đập vào màn hình phủ photpho gây ra phát quang: Mỏi mắt, chóng mặt do sự nhấp nháy, công kênh
 - LCD: Sử dụng công nghệ tinh thể lỏng: tốn ít năng lượng, gọn nhẹ: không có phóng xạ, có khả năng hiển thị từng điểm ảnh, cường độ sáng thấp nên ít nhấp nháy, giảm mỏi mắt
 - Màn hình cảm ứng
- Máy in
- Loa
- V.v.



4. Bộ nhớ

- Ngắn hạn: RAM
- Dài hạn: HD, CD



5. Xử lý thông tin

- Tốc độ xử lý : nếu tốc độ xử lý quá chậm => phản hồi thông tin chậm
- Để thiết kế hiệu quả, cần xem xét:
 - Giới hạn của khả năng tính toán
 - Giới hạn của kênh lưu trữ
 - Giới hạn về khả năng đồng bộ
 - Dung lượng mạng khi thiết kế các phần mềm hoạt động trên môi trường mạng



Kết chương

- Các nhân tố then chốt trong hệ tương tác người máy:
 - Con người
 - Máy tính
- Nghiên cứu khả năng, giới hạn, đặc tính của từng nhân tố
 - Kênh vào ra
 - Bộ nhớ
 - Xử lý thông tin
- Các lưu ý khi thiết kế tương tác để đảm bảo tính tiện dụng



Thảo luận

- Xác định các thiết bị đầu vào và đầu ra cho các hệ thống dưới đây.
 - Hệ thống xử lý văn bản cầm tay (portable word processor)
 - Hệ thống thông tin du lịch (Tourist Information system)
 - Hệ thống điều khiển lộ trình bay (air traffic control system)
- Đối với mỗi hệ thống, phân tích và so sánh và đánh giá tại sao các thiết bị truyền thống như: bàn phím, chuột, và màn hình CRT không phù hợp trong các hệ thống đó.

Portable word processors

- Input:
- Output:
- Ưu điểm:

- Nhược điểm



Tourist Information system

- Input:
- Output:
- Ưu điểm:

- Nhược điểm



Air traffic control system

- Input:
- Output:
- Ưu điểm:

- Nhược điểm






PHẦN III: THIẾT KẾ GIAO DIỆN TƯƠNG TÁC NGƯỜI DÙNG – MÁY TÍNH



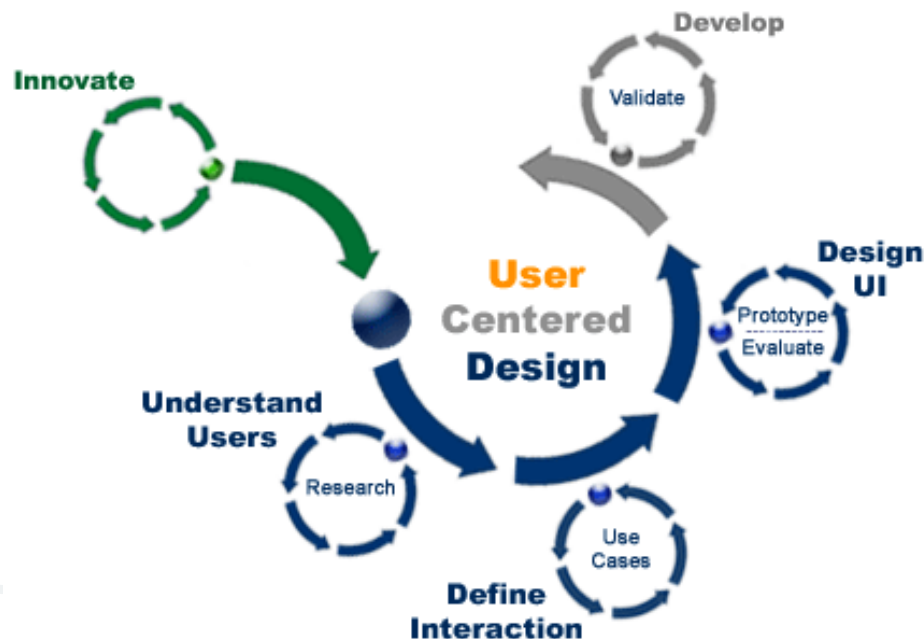


CHƯƠNG I. THIẾT KẾ LẤY NGƯỜI DÙNG LÀM TRUNG TÂM

- 
- I. Khái niệm
 - II. Nguyên tắc
 - III. Các phương pháp thiết kế lấy người dùng làm trung tâm
 - IV. Thiết kế hợp tác

Thiết kế lấy người dùng làm trung tâm (User-centered design)

- Đặt (yêu cầu của) người dùng vào trung tâm của quá trình thiết kế và phát triển
- Tiến hành thử nghiệm và đánh giá với người dùng
- Thiết kế tương tác
- Quy trình thiết kế là quy trình lặp



Quy trình thiết kế lấy người dùng làm trung tâm

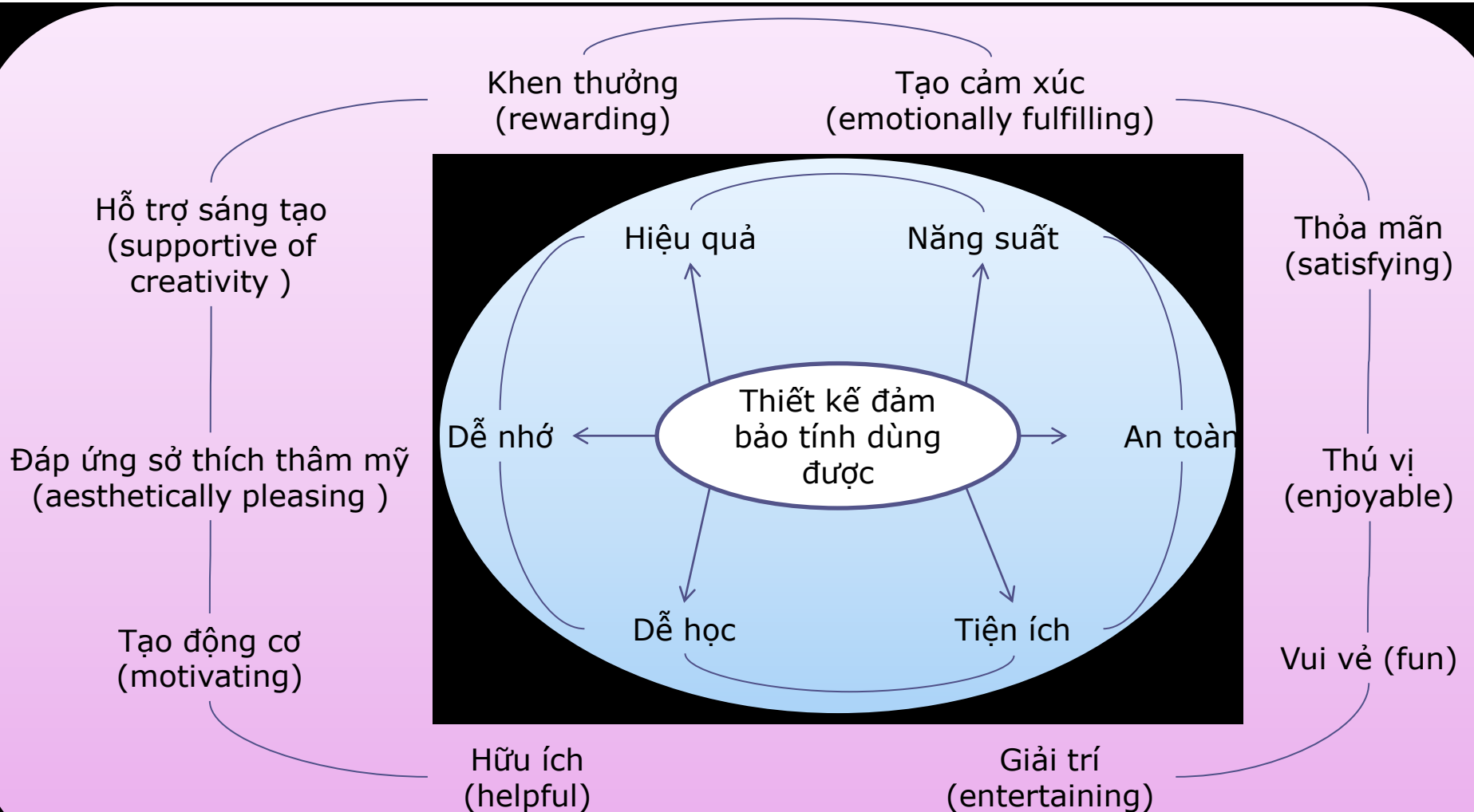
- Đặc tả yêu cầu
 - Người dùng là ai
 - Mục đích của họ là gì
 - Nhiệm vụ nào họ muốn hoàn thành
- Phân tích nhiệm vụ
 - Đặc trưng hóa các bước mà người dùng cần thực hiện
 - Xây dựng kịch bản đối với việc sử dụng hiện tại
 - Đưa quyết định hỗ trợ đối tượng người dùng và loại nhiệm vụ
- Thiết kế dựa trên các phân tích này
- Đánh giá
 - Thử nghiệm giao diện
 - Đánh giá trước khi cài đặt



CHƯƠNG I. THIẾT KẾ LẤY NGƯỜI DÙNG LÀM TRUNG TÂM

- 
- I. Khái niệm
 - II. Nguyên tắc**
 - III. Các phương pháp thiết kế lấy người dùng làm trung tâm

Mục tiêu





Tương thích

- Người dùng: Thiết kế phải thích hợp với nhu cầu của người dùng
- Nhiệm vụ: Thiết kế phải đảm bảo các luồng dữ liệu và chức năng thực hiện đúng nhiệm vụ đề ra, người dùng không phải cố gắng để thực hiện một nhiệm vụ đề ra
- Sản phẩm: Người dùng những hệ thống mới thường là những người dùng của hệ thống phiên bản trước đó. Vì thế sản phẩm phải được thiết kế sao cho các thói quen, mong đợi và sự hiểu biết của người dùng cũ vẫn có thể áp dụng được.



Thẩm mỹ và rõ ràng

Thẩm mỹ

- Một thiết kế có tính thẩm mỹ phải cuốn hút được người dùng.

Rõ ràng

- Giao diện phải rõ ràng về mặt hiển thị, thiết kế cũng như ngữ nghĩa



Dễ cấu hình và nhất quán

Cấu hình

- Cho phép dễ dàng cá nhân hóa và cung cấp khả năng cấu hình được.

Nhất quán

- Một hệ thống phải được nhìn thấy, phản ứng, thao tác theo cùng một cách trong cùng một ngữ cảnh.



Trực tiếp và hiệu quả

Trực tiếp

- Cung cấp cách thức hoàn thành nhiệm vụ một cách trực tiếp:

Hiệu quả

- Tối thiểu hóa chuyển động của tay, mắt và các hoạt động điều khiển khác
- Dự báo những mong muốn của người dùng và nhu cầu mỗi khi có thể.

Thân thiện và mềm dẻo

Thân thiện

- Sử dụng các khái niệm và ngôn ngữ gần gũi với người dùng
- Giao diện nên tự nhiên, bắt chước những mẫu hành vi của người dùng

Mềm dẻo

- Hệ thống phải nhạy với những nhu cầu khác nhau của người dùng, cho phép thực hiện yêu cầu ở các mức độ với một hiệu quả nào đó dựa trên
 - Hiểu biết và kỹ năng của người dùng
 - Kinh nghiệm của người dùng
 - Sở thích cá nhân
 - Thói quen
 - Điều kiện hiện tại

Bỏ qua lỗi và khôi phục trạng thái

Bỏ qua

- Dung thứ và bỏ qua một số lỗi chung không tránh khỏi của người dùng
- Dự báo lỗi nếu có thể
- Đề phòng những lỗi trầm trọng.
- Mỗi khi có lỗi xuất hiện, đưa thông báo và cách giải quyết

Khôi phục

- Một hệ thống phải cho phép
 - Hủy bỏ hoặc quay lại một số lệnh, thao tác tại thời điểm trước đó
 - Quay lại bước một số trước đó nếu có khó khăn xảy ra.
- Đảm bảo người dùng không bao giờ bị mất kết quả tại trạng thái hiện tại

Trách nhiệm và đơn giản

Trách nhiệm

- Hệ thống phải trả lời nhanh những yêu cầu của người dùng
- Cung cấp phản hồi tức thì đối với mọi hành động của người dùng thông qua
 - Hình ảnh
 - Văn bản
 - Âm thanh.

Đơn giản

- Cung cấp giao diện đơn giản nhất có thể.
- Một số cách thức để tạo ra sự đơn giản:
 - Sử dụng các menu dạng ẩn, chỉ xuất hiện khi cần thiết.
 - Hiển thị các chức năng chung và cần thiết trước tiên.
 - Cung cấp các mặc định.
 - Tạo các hành động chung, đơn giản.

Trong suốt, dễ hiểu, dễ học

Trong suốt

- Cho phép người dùng tập trung vào nhiệm vụ, tránh tập trung vào các cơ chế, cách thức hoạt động của giao diện.
 - Các công việc và việc nhắc nhở các công việc trong máy tính phải trong suốt đối với người dùng.
- Không được bắt người dùng nghĩ về các chi tiết kỹ thuật của hệ thống. Các suy nghĩ phải hướng đến nhiệm vụ.

Dễ hiểu, dễ học

- Một hệ thống phải dễ học và dễ hiểu. Người dùng phải biết:
 - What to look at
 - What to do
 - When to do it
 - Where to do it
 - Why to do it
 - How to do it
- Chuỗi hành động, đáp ứng, trình diễn và thông tin phải theo trật tự có khả năng dự đoán được.

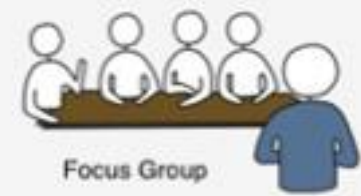


CHƯƠNG I. THIẾT KẾ LẤY NGƯỜI DÙNG LÀM TRUNG TÂM

- 
- I. Khái niệm
 - II. Nguyên tắc
 - III. Các phương pháp thiết kế lấy người dùng làm trung tâm**

Các phương pháp UCD phổ biến

- Các phương pháp này nên được kết hợp lại để thực hiện các tương tác với người dùng trong suốt quá trình tìm hiểu, phân tích, thiết kế, cài đặt thử nghiệm, vận hành và khai thác



Các phương pháp UCD phổ biến

Phương pháp	Chi phí	Kết quả đầu ra	Số lượng mẫu thu thập	Thời điểm sử dụng
Focus groups	Thấp	Không thống kê được	Ít	Thu thập yêu cầu người dùng
Usability testing	Cao	Thống kê được hoặc không	Ít	Thiết kế và đánh giá
Card Sorting	Cao	Thống kê được	Nhiều	Thiết kế
Participatory design	Thấp	Không thống kê được	Ít	Thiết kế
Questionnaires	Thấp	Thống kê được	Nhiều	Thu thập yêu cầu người dùng và đánh giá
Interviews	Cao	Không thống kê được	Ít	Thu thập yêu cầu người dùng và đánh giá

1. Focus groups - nhóm tập trung





Focus groups - nhóm tập trung

- What are they ?
 - Một nhóm người dùng được mời đến để cùng chia sẻ suy nghĩ, cảm nhận, thái độ, ý kiến về một chủ đề nào đó
- When to use ?
 - Kết quả thảo luận được sử dụng như là đầu vào thiết kế.
 - Thông thường kết quả này thường là các dữ liệu phi thống kê, là phương tiện để có được thông tin về một lĩnh vực, chủ đề (e.g. what peoples' tasks involve).
- Issues ?
 - Cần thiết phải có một nhóm trưởng có kinh nghiệm và một nhà phân tích để việc thảo luận có hiệu quả.



Câu hỏi đặt ra

- Ai được tham gia?
- Tại sao cần tham gia?
- Có bao nhiêu người trong 1 nhóm?
- Người tham gia phải làm gì ?
- Ai là moderator của nhóm ? Vai trò, nhiệm vụ cụ thể ?
- Mỗi phiên làm việc của nhóm dài bao lâu?
- Người tham gia có bị ghi âm hay ghi hình hay không ? Tại sao ?
- Những gì người tham gia nói có được giữ bí mật không?
- Ai có quyền truy cập vào các dữ liệu đã thu thập được?

2. Usability testing - Kiểm thử tính dùng được



Usability testing - Kiểm thử tính dùng được

- Đánh giá một vấn đề gì đó thông qua việc thu thập dữ liệu từ những người sử dụng nó
- Một người được mời để tham gia một phiên kiểm thử trong đó họ đòi hỏi để thực hiện một số nhiệm vụ
- Một người điều khiển sẽ ghi chép lại tất cả những khó khăn mà người dùng thử gặp phải
- Người dùng được hỏi họ đang làm gì và vì sao
- Đo thời gian người dùng hoàn thành một tác nhiệm
- Thông thường có hai chuyên viên làm việc cùng với người dùng thử: một người điều khiển, một người ghi chép

Khi nào sử dụng phương pháp này:

Được sử dụng làm đầu vào của quá trình thiết kế hoặc lúc kiểm tra mẫu thử hoặc kết thúc dự án

Là cách thức tốt nhất để tìm ra những khó khăn trong việc sử dụng sản phẩm

Có thể đưa ra những thông tin mang tính thông kê hoặc thông tin phi thống kê



Câu hỏi đặt ra

- Ai phải tham gia vào quá trình kiểm thử?
- Cần thực hiện bao nhiêu phiên kiểm thử?
- Có cần ghi hình các phiên kiểm thử không?

Ví dụ: Microsoft làm thế nào để tiếp cận người dùng ? (1995)

- Nghiên cứu về người dùng
- Mỗi khi một tính năng mới được phát triển, các thử nghiệm về tính dùng được sẽ được triển khai tại PTN
- Nhóm 10 người bất kỳ sẽ được đưa vào để thử nghiệm. Các hành vi sẽ được quan sát và hiệu năng sẽ được ghi lại
- Dữ liệu sẽ được phân tích và tìm ra các phản hồi để phát triển tiếp
- Ví dụ Office 4.0 được thử nghiệm sau 8000h thử nghiệm tính tiện dụng
- Mỗi khi sản phẩm hoàn thiện, nó sẽ được dùng thử bởi các thành viên của Microsoft
- Luôn có trung tâm chăm sóc khách hàng ghi lại các phản hồi của khách hàng.

3. Card Sorting



Card sorting – sắp xếp các phiếu đánh giá

- Nhóm chuyên gia, người dùng thử không có kinh nghiệm trong thiết kế được hướng dẫn để tạo ra một cây phân loại
- Là một hướng tiếp cận tốt để thiết kế kiến trúc thông tin, các công việc, cấu trúc menu, hay đường dẫn định hướng đến trên một trang websites nào đó
- Hoạt động có thể được tổ chức theo nhóm (focus groups) hay tiến hành với từng cá nhân
- Các khái niệm đầu tiên sẽ được xác định và viết trên các thẻ có đánh số (post-it notes)
- Những người tham gia được yêu cầu tổ chức nó thành một cấu trúc cây

Khi nào sử dụng phương pháp này:

Thường được sử dụng như đầu vào cho thiết kế. Là cách thức tốt để phân loại nội dung của một site và đưa ra kiến trúc về mặt thông tin

Có thể được sử dụng để sinh các dữ liệu thống kê

4. Participatory design – cùng tham gia thiết kế

- Không chỉ đòi hỏi ý kiến của người dùng về thiết kế mà còn yêu cầu họ tham gia vào việc thiết kế cũng như quá trình ra quyết định

Khi nào sử dụng phương pháp này:

Được sử dụng đối với các dự án nhỏ nhằm đưa ra các mẫu thử để đưa vào quá trình thiết kế tổng thể

Các dự án cho phép người dùng tham gia vào quá trình thiết kế và ra quyết định là rất hiếm



5. Questionnaires – lập bảng câu hỏi

- Là một phương tiện để hỏi người dùng về các vấn đề nào đó, dựa trên một tập các câu hỏi đã được định nghĩa sẵn
- Là một cách thức tốt để tạo thông tin thống kê.

Khi nào sử dụng phương pháp này:

Có thể cho phép người sử dụng từ xa tiến hành tham gia thông qua các trang web

Cho phép số lượng mẫu người dùng thử lớn

Thường được quản lý thông qua các phương tiện điện tử

6. Interviews – phỏng vấn

- Một người phỏng vấn một người khác tại một thời điểm.
- Ưu điểm: quan điểm cá nhân của người dùng thử sẽ được khai thác và ghi nhận
- Những hiểu lầm giữa người phỏng vấn và người được phỏng vấn được nhanh chóng sửa lỗi
- Đầu ra: có thể là những thông tin phi thống kê, những ý kiến này sẽ được nghiên cứu phân tích bởi các chuyên viên có kinh nghiệm.

Khi nào sử dụng phương pháp này:

Thường diễn ra trước quá trình thiết kế nhằm thu thập các thông tin, những tri thức về lĩnh vực hoạt động hay những yêu cầu cụ thể

Ví dụ về UCD: Pizza ordering system

- Thiết kế website đặt pizza trực tuyến
- Đề xuất các tính năng mới
 - Phiếu giảm giá trực tuyến
 - Cho phép đặt hàng trước
 - Khuyến nghị cho khách hàng
 - Liên kết để đánh giá
 - So sánh giá với các tiệm pizza khác





Xác định yêu cầu

- Nhiệm vụ của bạn: Chỉ ra
 - Việc đặt mua pizza qua mạng hiện tại như thế nào
 - Họ thích đặt mua trực tuyến ra sao
 - Tính năng nào sẽ có ý nghĩa khi nào
 - Xem xét đến các yêu cầu của chủ tiệm và nhân viên
- Đánh giá nhu cầu
- Các kỹ thuật sử dụng
 - Quan sát
 - Phỏng vấn
 - Nghiên cứu các thiết kế đã thành công



Kỹ thuật quan sát

- Đến thăm một cửa hàng bán pizza
- Quan sát xem mọi người thực hiện việc mua pizza như thế nào
- Quan sát xem người thu ngân làm việc như thế nào, có thể phỏng vấn họ



Phỏng vấn

- Chuẩn bị một danh sách các câu hỏi xem mọi người thực hiện order như thế nào và điều gì họ mong muốn ở một hệ thống đặt pizza tự động
- Phỏng vấn ít nhất 3 người xem họ mong muốn hệ thống đặt hàng trực tuyến sẽ hoạt động như thế nào
 - Xác định các yêu cầu, sở thích và thái độ của người phỏng vấn về hệ thống đặt hàng trực tuyến.
- Hỏi họ thêm xem họ có cần hỗ trợ trực tuyến, hay họ thích sử dụng hệ thống đặt hàng qua điện thoại
 - Các yêu cầu khác từ phía người dùng: Không có tín hiệu bận, giá cả rẻ hơn, so sánh giá, dịch vụ chuyển phát nhanh, phương pháp thanh toán



Câu hỏi phỏng vấn

- Bạn thích tự lựa chọn nhân bánh/ đế bánh không hay lựa chọn loại bánh mình thích trong số các loại bánh do cửa hàng cung cấp?



Tóm tắt kết quả phỏng vấn

- Tự chọn nhân bánh/ đế bánh (A)(B)(C)

Tìm hiểu các thiết kế tương tự

- Xem xét ít nhất một website bán thức ăn trực tuyến: làm thế nào họ kiểm soát các hoạt động





Các công việc tiếp theo

- Trả lời các câu hỏi đánh giá
- Tìm hiểu các nhiệm vụ cơ bản đang được hỗ trợ tại các tiệm pizza và đặt hàng qua điện thoại
- Tạo một bảng chỉ ra
 - Loại người dùng
 - Các nhiệm vụ
 - Ước lượng về tần suất của mỗi nhiệm vụ
- Quyết định các tác nhiệm mới với giao diện đặt hàng mới.
 - Dựa trên ghi chép từ phỏng vấn.



Các câu hỏi đánh giá



Phân tích nhiệm vụ

- Phân tích và cụ thể hóa điều gì sẽ xảy ra khi người dùng thực hiện một tác nhiệm nào đó
- Công cụ:
 - Bảng ai làm gì
 - Who x What
 - Bảng chuỗi các nhiệm vụ
 - Lược đồ, sơ đồ chuyển đổi
 - Bảng video mô tả kịch bản

Ví dụ: Tạo bảng ai làm gì

Job Title	Query by Patient	Update Data	Query across Patients	Add Attributes
Nurse	4	4		
Physician	4	4		
Supervisor	4	4	4	
Appointment administrator	4			
Clinical researcher			4	
Database programmer			4	4

Ví dụ: thêm thông tin về khối lượng công việc mỗi người thực hiện

Job Title	Query by Patient	Update Data	Query across Patients	Add Attributes
Nurse	14%	11%		
Physician	6%	4%		
Supervisor	1%	1%	4%	
Appointment administrator	26%			
Clinical researcher			8%	
Database programmer			2%	2%



Thiết kế

- Phác thảo thiết kế ban đầu
 - Giải thích cách thức hoạt động của hệ thống.
 - Dùng ngôn ngữ tự nhiên và/hoặc biểu đồ để chỉ ra thứ tự các bước được phép thực hiện trong hệ thống.
 - Phác thảo thô ít nhất 3 trong số các màn hình quan trọng.
 - Kiểm thử (Walk through) ít nhất một kịch bản sử dụng và chỉ ra làm thế nào thực hiện kịch bản đó thông qua giao diện.



Ví dụ: online ordering pizza

- Dựa trên kết quả phỏng vấn người bán hàng, xác định được 5 nhiệm vụ chính của quy trình bán hàng hiện tại:



Ví dụ: online ordering pizza

- Trong số các chức năng mới đề xuất, dựa vào phỏng vấn nhân viên, khách hàng và quan sát hệ thống thực để lựa chọn 5 chức năng đưa vào giao diện của hệ thống online, loại bỏ các chức năng còn lại



Ví dụ: online pizza ordering

- Các kịch bản sau được tạo ra trên cơ sở phân loại người dùng theo tần suất mua hàng, thời gian giao hàng và phương pháp đặt mua. Trong hệ thống mới, có thể đặt mua trực tiếp hoặc đặt mua qua mạng



Kịch bản 1: người mua hàng thường xuyên,
đặt mua tại nhà, cần giao hàng ngay



Kịch bản 2: đặt mua từ văn phòng, thời gian giao hàng theo hẹn trước, lần đầu đặt mua



Kịch bản 3: Đặt mua tại chỗ, giao hàng sau khi đặt hàng 2h, lần đầu đặt mua

- **The new online pizza ordering system for PRU works in the following steps;**

- *(a..e): related interface images following the flow chart.

-
-
-



CHƯƠNG II. CÁC KỸ THUẬT THIẾT KẾ GIAO TIẾP NGƯỜI DÙNG – MÁY TÍNH





Tổng kết bài học

- Tổng quan về giao diện tương tác người dùng máy tính, các nguyên lý thiết kế
- Hướng tiếp cận thiết kế tương tác lấy người dùng làm trung tâm
- Phân tích một ví dụ về UCD
- Buổi sau: THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG



CHƯƠNG II. THIẾT KẾ HỢP TÁC





IV. Thiết kế hợp tác

- Thiết kế theo ngữ cảnh công việc, xem người sử dụng không chỉ như là một đối tượng thí nghiệm mà còn như một thành viên của nhóm thiết kế.
- Nhằm mục đích cải tiến các yêu cầu hệ thống thông qua một quy trình thiết kế mang tính lặp trong đó người sử dụng tham gia một cách tích cực vào quy trình thiết kế.



Đặc điểm của thiết kế hợp tác

- Cải tiến môi trường làm việc và nhiệm vụ thông qua việc giới thiệu thiết kế.
 - Làm cho thiết kế và đánh giá mang tính hướng ngữ cảnh hoặc công việc hơn là hướng hệ thống.
- Đặc trưng bởi sự hợp tác: người sử dụng tham gia vào nhóm thiết kế và có thể đóng góp ở mọi giai đoạn của quá trình thiết kế.
- Cách tiếp cận của thiết kế hợp tác mang tính lặp: thiết kế có thể được đánh giá và chỉnh sửa lại ở mỗi giai đoạn.



Quy trình thiết kế hợp tác

- Sử dụng một số phương pháp để giúp cho việc trao đổi thông tin giữa người sử dụng và người thiết kế
 - Brainstorming (Phương pháp trí tuệ nhóm):
 - Mọi thành viên tham gia thiết kế đều đưa ra ý tưởng
 - Mọi thông tin đều được ghi lại trung thực, không chỉnh sửa
 - Chọn lọc ý tưởng bằng nhiều kỹ thuật khác nhau.
 - Storyboarding (Phương pháp xây dựng bảng tình tiết)
 - Người dùng là tất cả những người có quyền lợi/nghĩa vụ liên quan
 - Sử dụng họ để mô tả các hoạt động của người dùng và đánh giá các tác động của thiết kế

Quy trình thiết kế hợp tác

- Sử dụng một số phương pháp để giúp cho việc trao đổi thông tin giữa người sử dụng và người thiết kế
 - Workshops (hội thảo):
 - Chất vấn lẫn nhau nhằm mục đích cho các bên tham gia có thể hiểu hơn về ngữ cảnh thiết kế từ quan điểm của mỗi thành viên.
 - Tạo ra một nền tảng chung giữa người sử dụng và người thiết kế và thiết lập cơ sở cho thiết kế.
 - Pencil and paper exercises (Xây dựng phác thảo):
 - Phác thảo các nhiệm vụ điển hình của hệ thống: chỉ rõ sự khác nhau giữa các yêu cầu của người sử dụng và thiết kế thực tế.
 - Cung cấp một kỹ thuật đơn giản và ít tốn kém cho việc đánh giá các mô hình ban đầu.



CHƯƠNG III. THIẾT KẾ MÀN HÌNH

- 
- I. Quy trình thiết kế màn hình
 - II. Nguyên tắc thiết kế màn hình tốt



I. Quy trình thiết kế màn hình

- [Step 1](#): Know Your User or Client
- [Step 2](#): Understand the Business Function
- [Step 3](#): Understand the Principles of Good Screen Design
- [Step 4](#): Develop System Menus and Navigation Schemes
- [Step 5](#): Select the Proper Kinds of Windows
- [Step 6](#): Select the Proper Device-Based Controls
- [Step 7](#): Choose the Proper Screen-Based Controls
- [Step 8](#): Write Clear Text and Messages
- [Step 9](#): Provide Effective Feedback and Guidance and Assistance
- [Step 10](#): Provide Effective Internationalization and Accessibility
- [Step 11](#): Create Meaningful Graphics, Icons and Images
- [Step 12](#): Choose the Proper Colors
- [Step 13](#): Organize and Layout Windows and Pages
- [Step 14](#): Test, Test, and Retest

II. Nguyên tắc thiết kế màn hình tốt

- Màn hình được thiết kế tốt nếu:
 - Phản ánh được năng lực, nhu cầu và nhiệm vụ của người dùng
 - Phù hợp với các ràng buộc vật lý của thiết bị hiện thị.
 - Sử dụng hiệu quả khả năng của các phần mềm điều khiển
 - Đạt mục tiêu nghiệp vụ của hệ thống.
- Nguyên tắc thiết kế màn hình tốt ???
 - Số lượng tin cần giới thiệu trên màn hình:
 - Cách tổ chức màn hình và phân tách các phần thông tin
 - Ngôn ngữ
 - Sự nhất quán giữa các màn hình



1. Màn hình và phần tử thông tin

- Các phần tử thông tin trên màn hình đều phải có ý nghĩa đối với người dùng: trợ giúp thực hiện nhiệm vụ
 - Điều khiển
 - Văn bản
 - Tổ chức màn hình
 - Nhấn mạnh
 - Màu sắc
 - Đồ họa
 - Hoạt họa
 - Thông điệp
 - Thông tin phản hồi

2. Nguyên tắc sắp xếp thứ tự thông tin trên màn hình

- Phân chia thông tin thành các phần logic, có ý nghĩa và dễ cảm nhận
- Tổ chức thông tin theo các cấp độ quan hệ của chúng.
- Sắp xếp thông tin theo kỳ vọng và nhu cầu người dùng
- Tạo các nhóm thông tin thỏa mãn các thứ tự sắp xếp phổ biến:
 - Quy ước
 - Trình tự sử dụng
 - Tần suất sử dụng
 - Chức năng
 - Mức độ quan trọng
 - Mức độ tổng quát.
- Các thông tin cần so sánh phải xuất hiện cùng lúc
- Chỉ các thông tin liên quan đến nhiệm vụ hay nhu cầu của người dùng mới xuất hiện trên màn hình

3. Luồng duyệt tin trên màn hình

- Mắt người thường dịch chuyển từ trái sang phải, từ trên xuống dưới và theo chiều kim đồng hồ

Thông tin được tìm thấy nhanh nhất	
	Thông tin được tìm thấy chậm nhất

3. Luồng duyệt tin trên màn hình

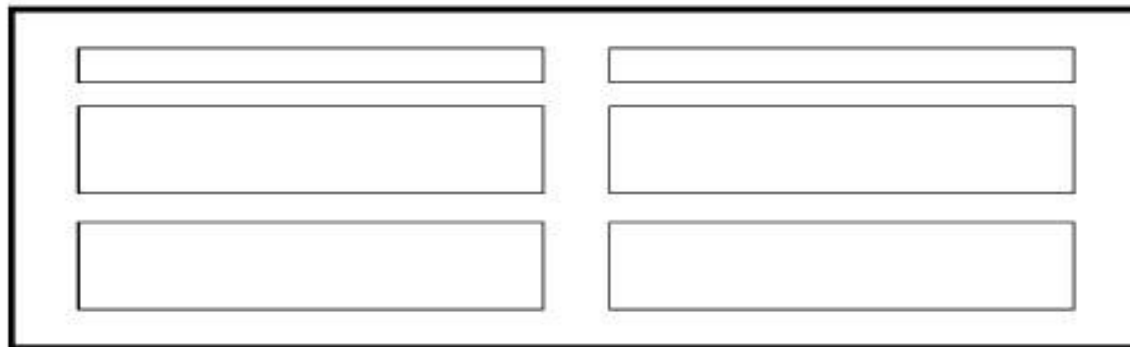
Mắt người thường dịch chuyển từ trái sang phải, từ trên xuống dưới và theo chiều kim đồng hồ

Thông tin được tìm thấy nhanh nhất	
	Thông tin được tìm thấy chậm nhất

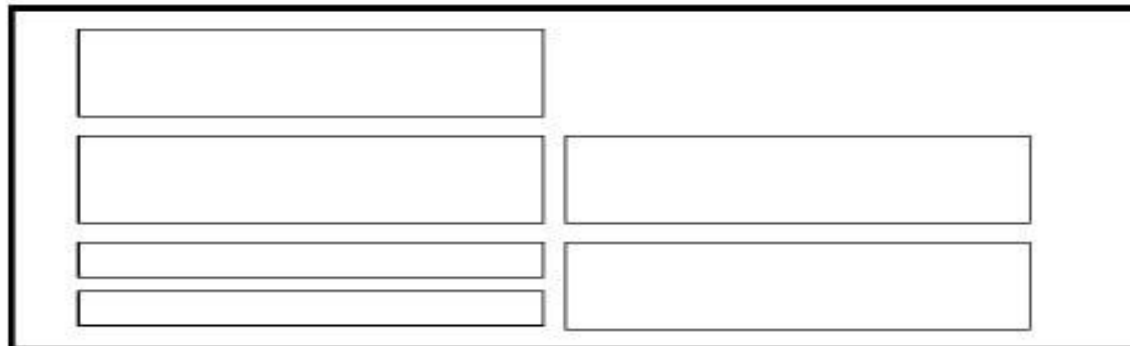
- Tổ chức màn hình sao cho :
 - Mắt người dùng duyệt qua các thông tin một các nhịp nhàng, có định hướng
 - Tôn trọng cách dịch chuyển tự nhiên của mắt
 - Tối thiểu hóa khoảng cách dịch chuyển giữa con trỏ và mắt
- Đặt các phần tử thông tin hay điều khiển quan trọng nhất, hay xuất hiện nhất vào góc trên bên trái màn hình
- Duy trì luồng thông tin từ trái sang phải, từ trên xuống dưới
- Hỗ trợ duyệt tin :
 - Gióng hàng các phần tử
 - Nhóm các phần tử
 - Sử dụng khung viền
- Tập trung và nhấn mạnh vào các phần tử: quan trọng, thứ cấp, ngoại vi
- Dùng phím tab để dịch chuyển theo thứ tự logic của các thông tin hiện thị
- Đặt các nút lệnh vào cuối dãy dịch chuyển của phím Tab
- Khi một nhóm các thông tin bị hiện thị trên vài màn hình khác nhau, cần cung cấp điểm ngắt thông tin rõ ràng trong luồng thông tin

Cân bằng

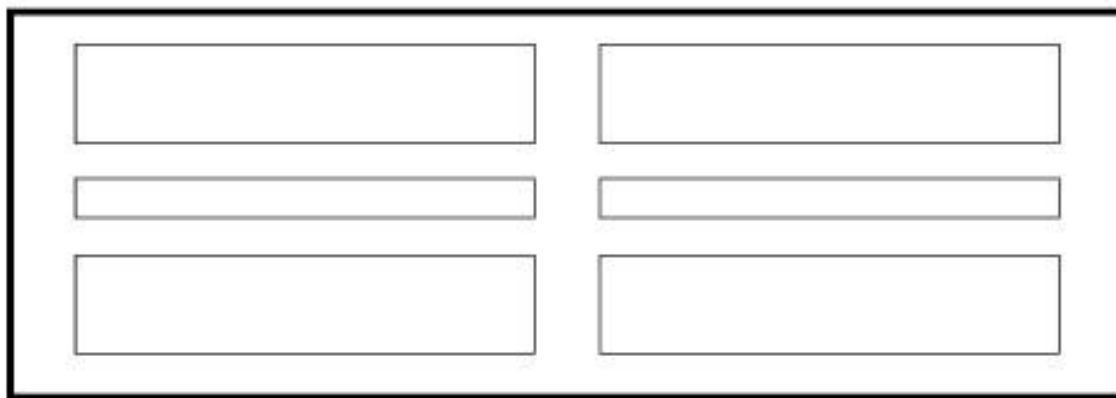
- Tạo sự cân bằng màn hình bằng cách xây dựng lưới cho các thành phần thông tin hiển thị trên màn hình: cùng độ dài, cùng độ rộng



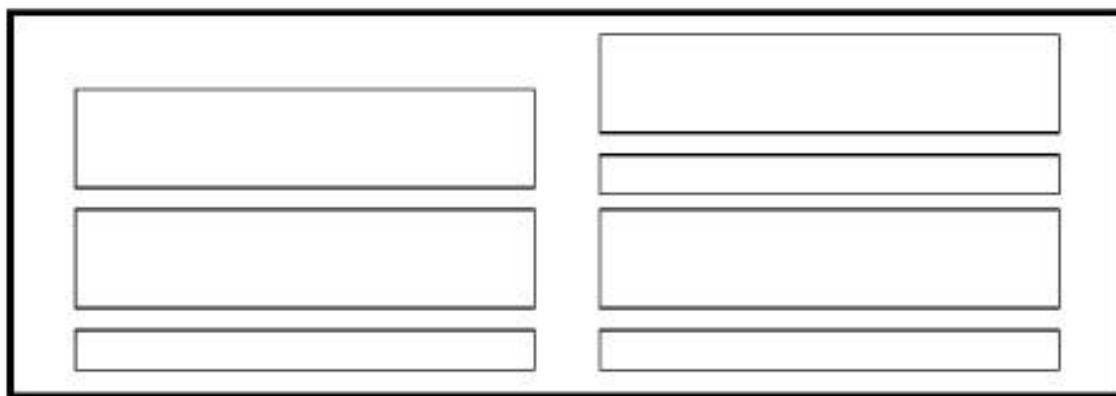
Balance



Đối xứng

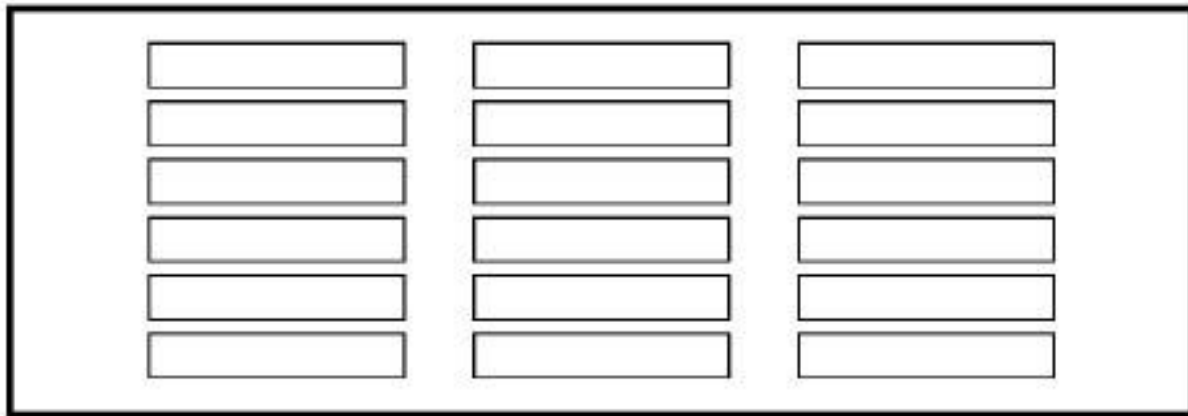


Symmetry

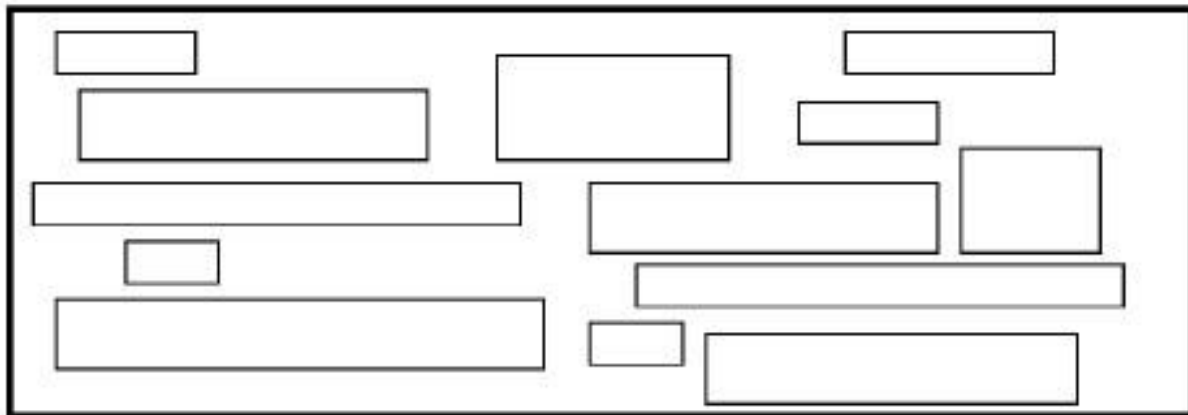


Asymmetry

Đều đặn



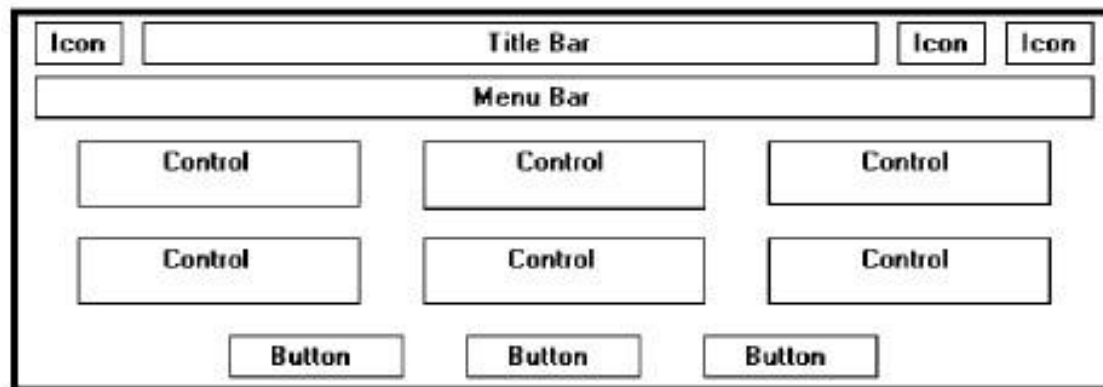
Regularity



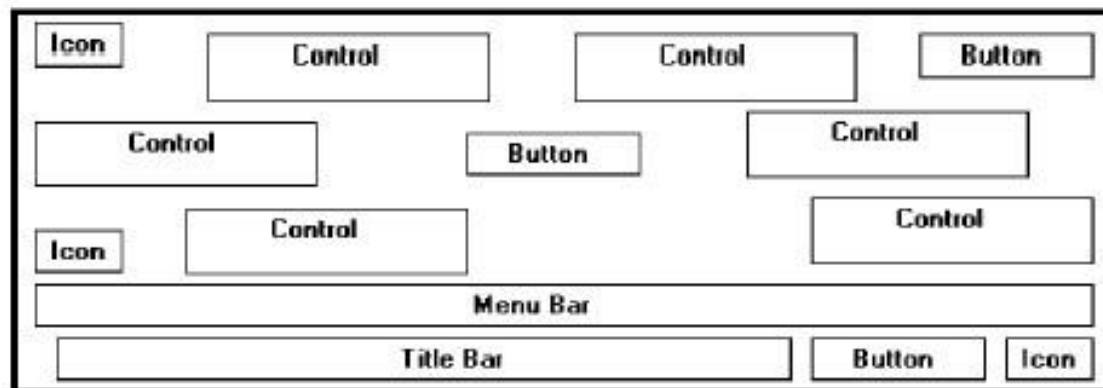
Irregularity

Nguyên tắc thiết kế luồng duyệt tin trên màn hình: Dễ dự đoán

- Tạo khả năng dự đoán theo các thứ tự đã thỏa thuận trước đó



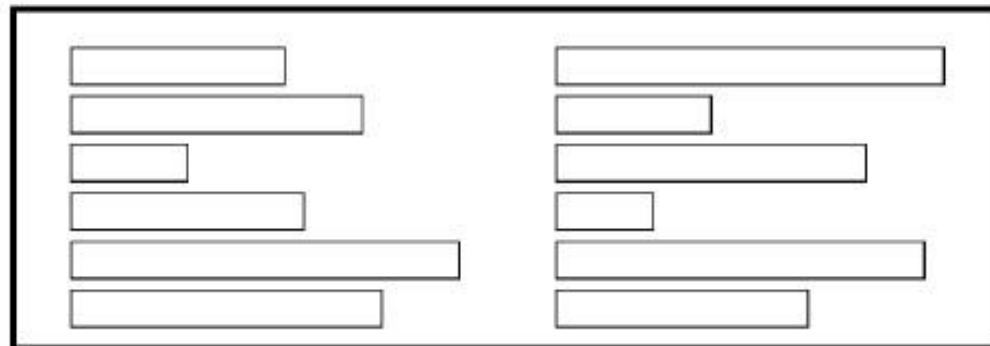
Predictability



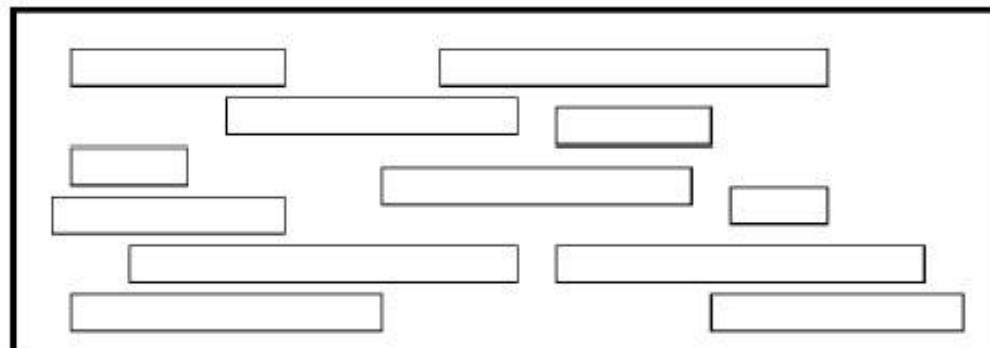
Spontaneity

Tuần tự

- Sắp xếp các yếu tố hướng dẫn mắt nhìn các thông tin trên màn hình một cách hợp lý, rõ ràng, nhịp nhàng và hiệu quả.



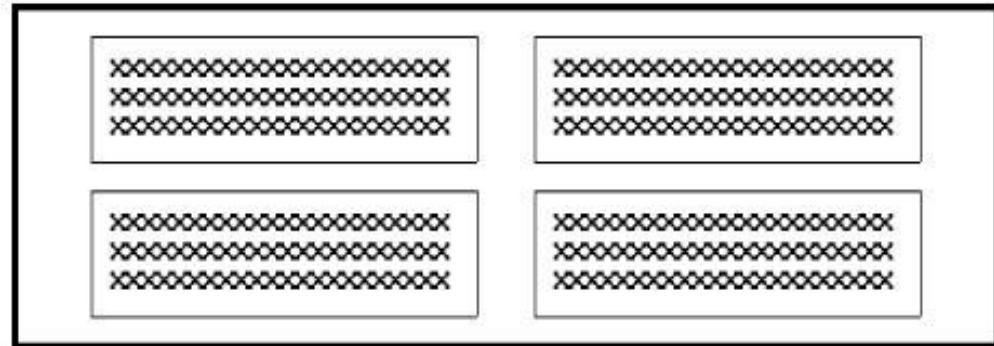
Sequentiality



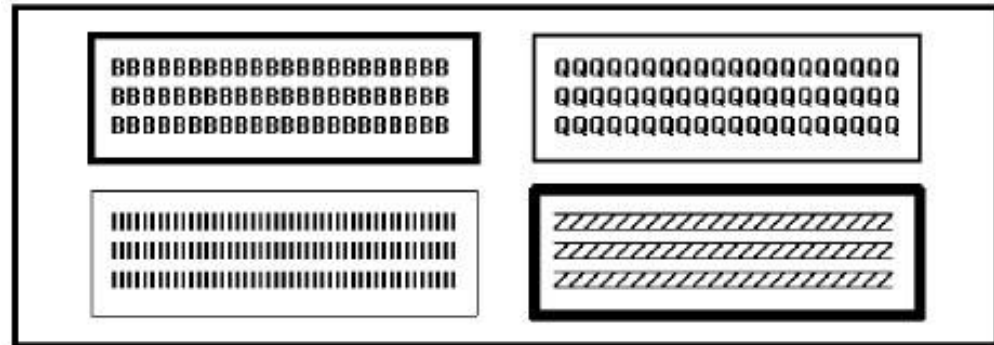
Randomness

Kinh tế

- Chỉ sử dụng các kỹ thuật hiện thị (như màu sắc, phông chữ..) đủ để làm nổi bật thông điệp gửi đến người dùng, không hơn.



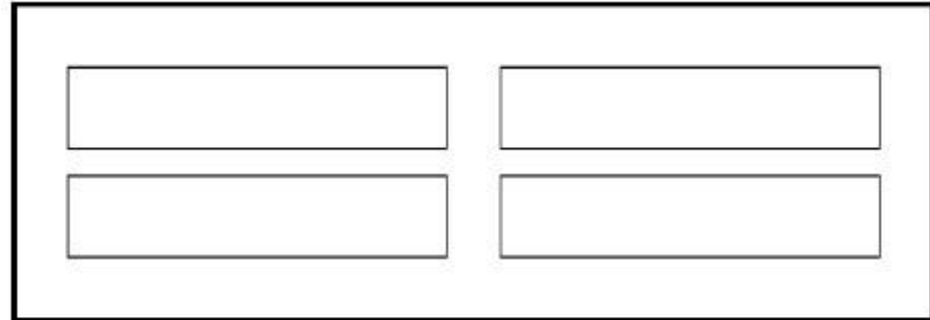
Economy



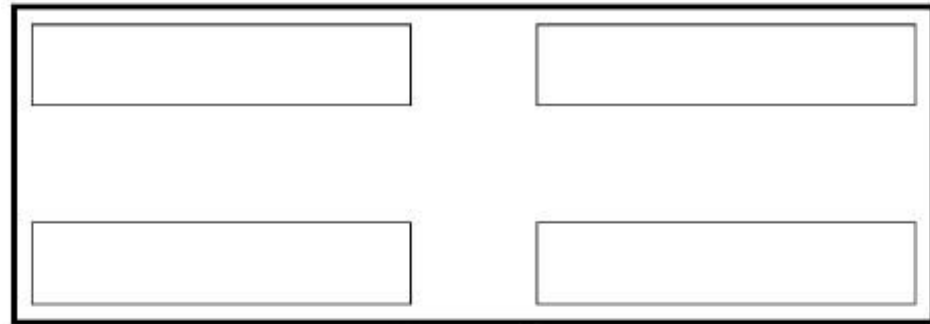
Intricacy

Thống nhất và phân mảnh

- Thống nhất: Tất cả các phần tử thông tin hiện thị trên màn hình đều là các mảnh ghép ăn khớp với nhau của một bức tranh toàn cảnh duy nhất.
- Phân mảnh: mỗi phần tử thông tin đều giữ lại đặc tính của riêng mình
 - Sử dụng kích thước, hình dạng, màu sắc tương tự cho các thông tin liên quan.
 - Phân biệt không gian giữa các nhóm tin (khoảng cách, biên giới)



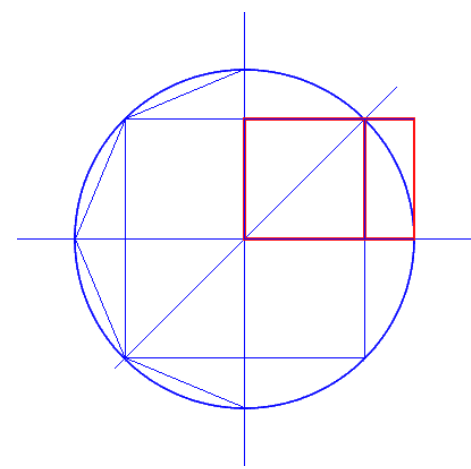
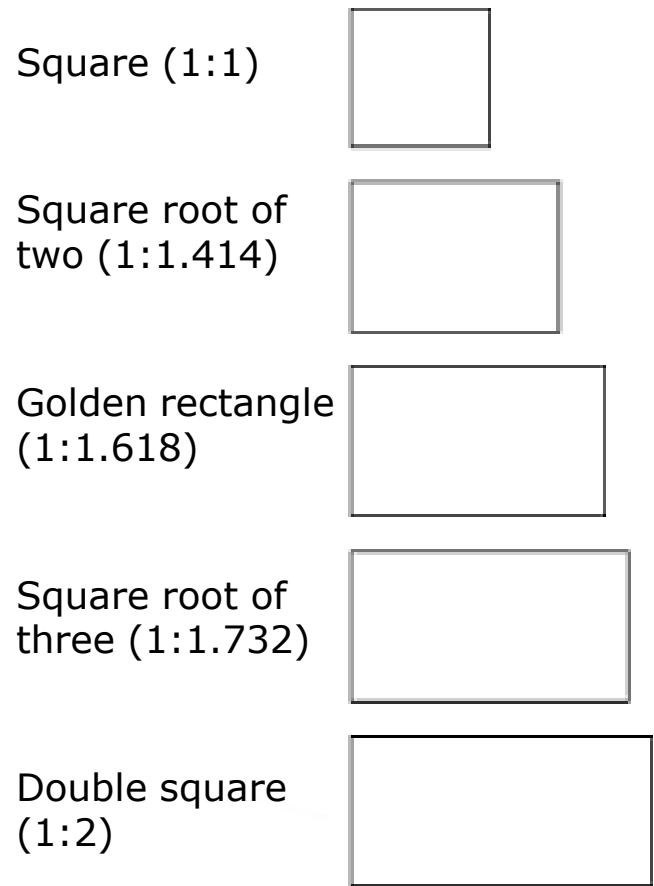
Unity



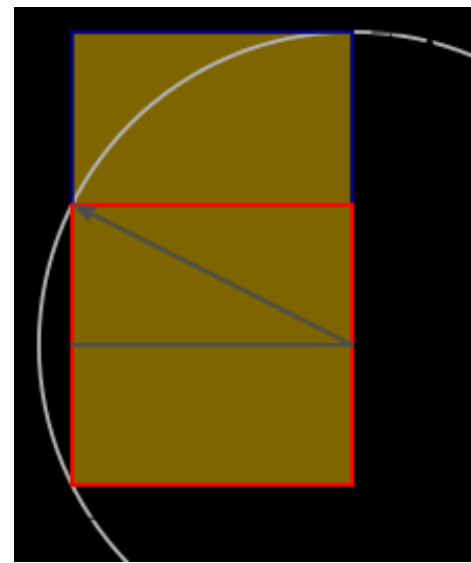
Fragmentation

Tỉ lệ

- Tạo các cửa sổ và các nhóm tin với tỉ lệ cân đối, đảm bảo yếu tố thẩm mỹ



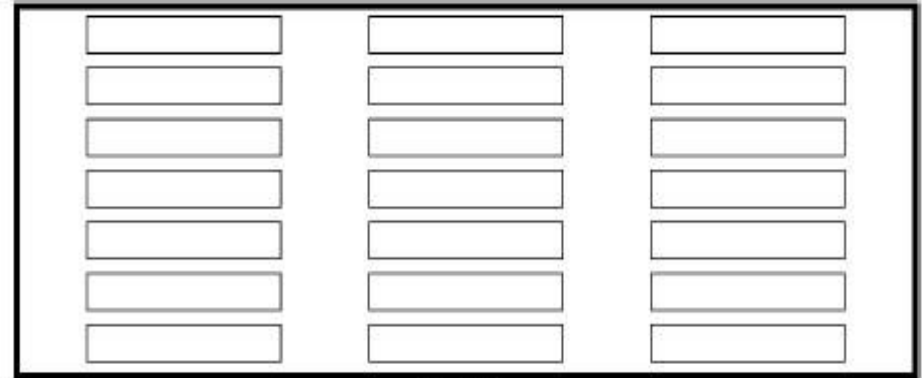
Square root of two
(1:1.414)



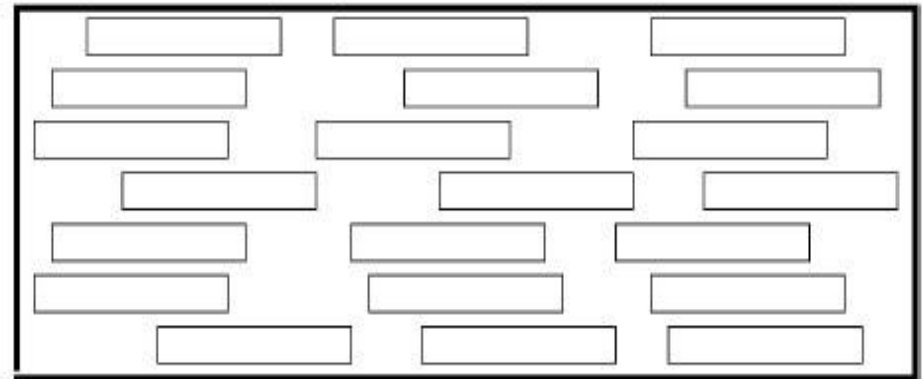
Golden rectangle
(1:1.618)

Đơn giản

- Tối ưu số phần tử thông tin trên màn hình, để đảm bảo nội dung từng phần tử đều được hiển thị rõ ràng
- Giảm thiểu số điểm giống hàng ngang – dọc



Simplicity



Complexity

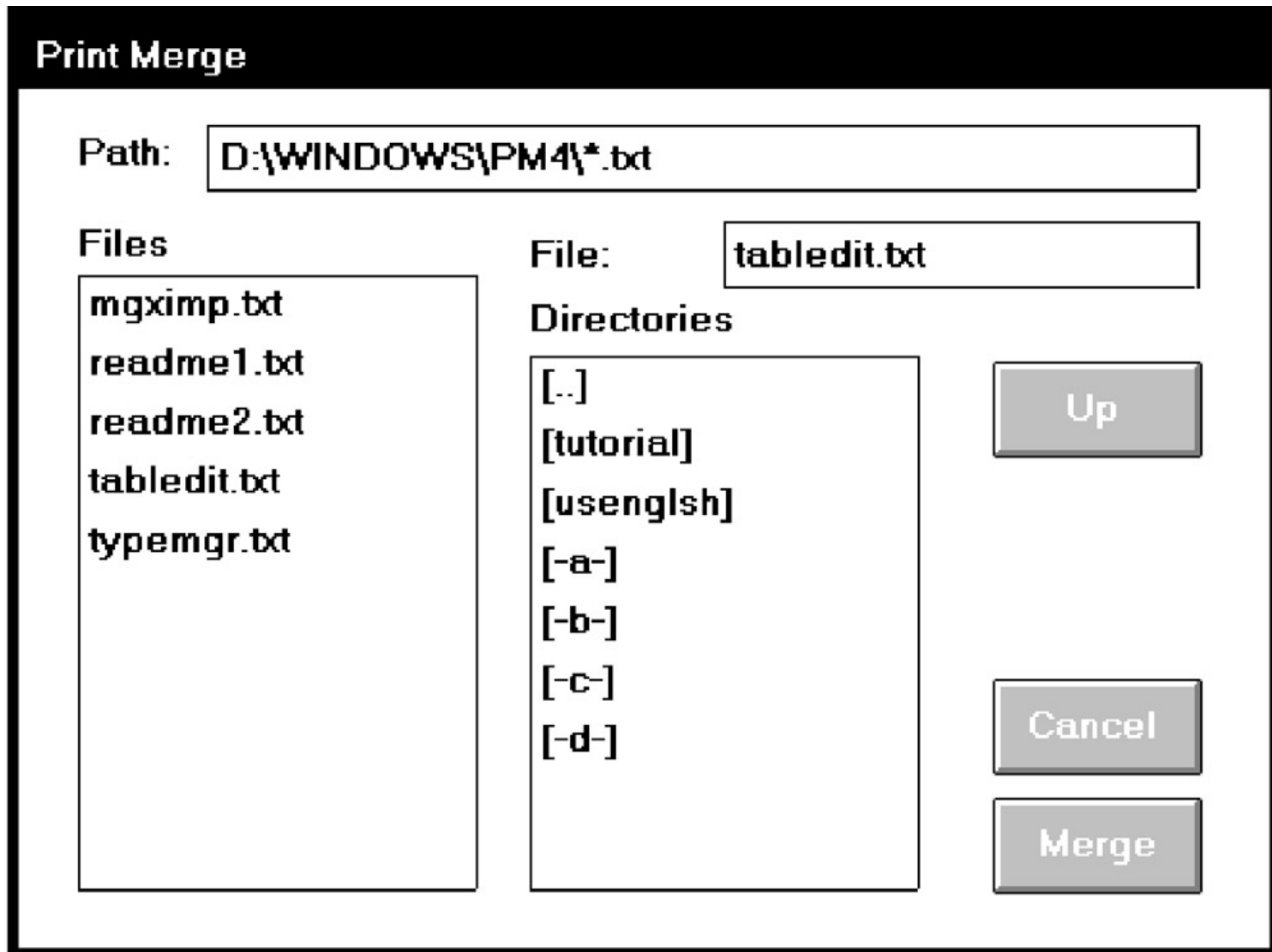
Nhóm thông tin

- Các thông tin liên quan được nhóm với nhau, có tiêu đề rõ ràng, có khoảng cách và đường viền hoặc màu nền hợp lý.

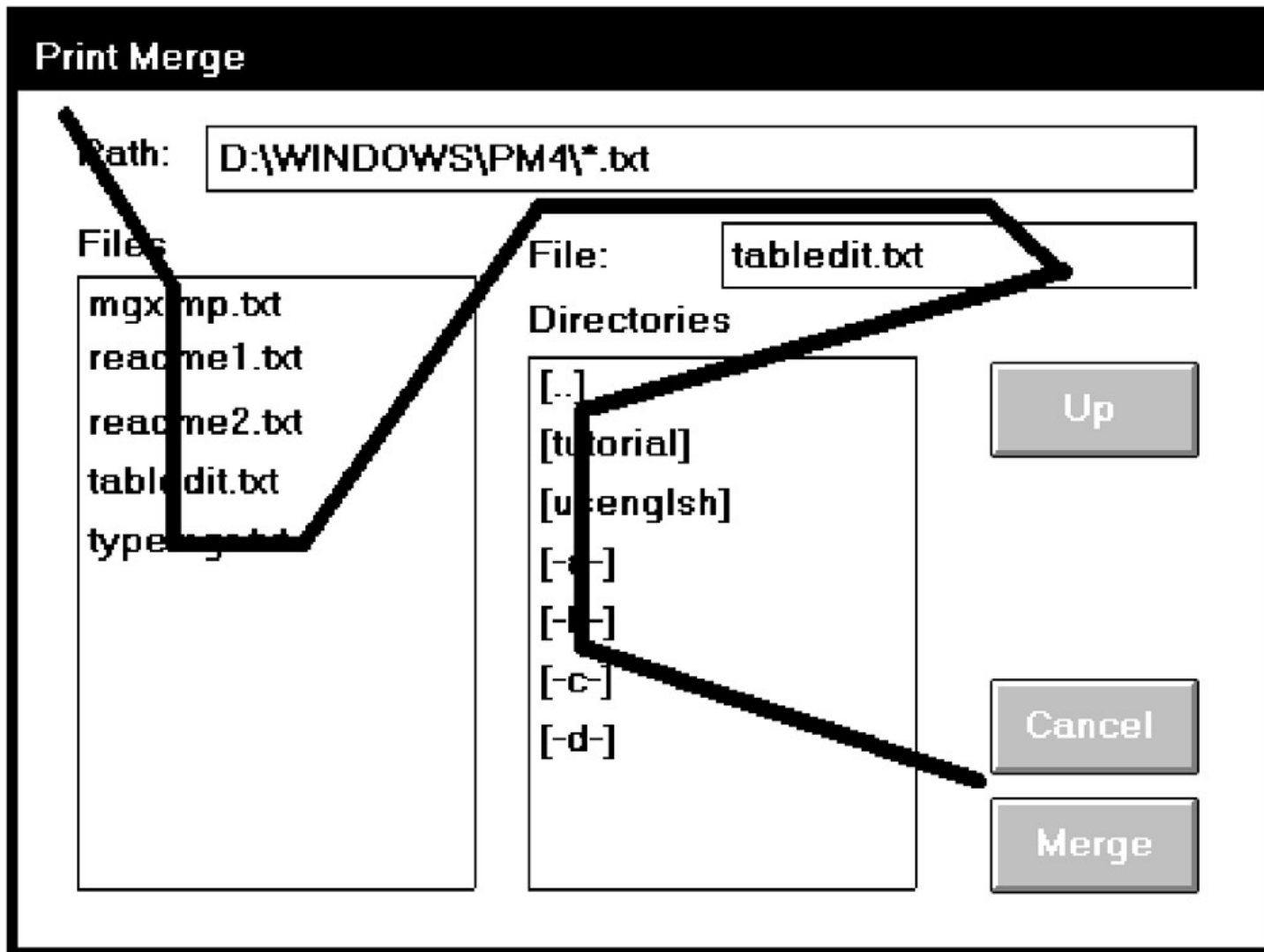
TEST RESULTS	SUMMARY: GROUND
GROUND, FAULT T-G	
3 TERMINAL DC RESISTANCE	
>	3500.00 K OHMS T-R
=	14.21 K OHMS T-R
>	3500.00 K OHMS R-G
3 TERMINAL DC VOLTAGE	
=	0.00 VOLTS T-G
=	0.00 VOLTS R-G
VALID AC SIGNATURE	
3 TERMINAL AC RESISTANCE	
=	8.82 K OHMS T-R
=	14.17 K OHMS T-R
=	628.52 K OHMS R-G
LONGITUDINAL BALANCE POOR	
=	39 DBB
COULD NOT COUNT RINGERS DUE TO LOW RESISTANCE	
VALID LINE CKT CONFIGURATION	
CAN DRAW AND BREAK DIAL TONE	

TIP GROUND 14 K		
DC RESISTANCE	DC VOLTAGE	AC SIGNATURE
3500 K T-R 14 K T-G 3500 K R-G	0 V T-G 0 V R-G	9 K T-R 14 K T-G 629 K R-G
BALANCE		CENTRAL OFFICE
39 DB		VALID LINE CKT DIAL TONE OK

3. Một số ví dụ về màn hình



Một số ví dụ về màn hình





Một số ví dụ về màn hình

PAGE SETUP

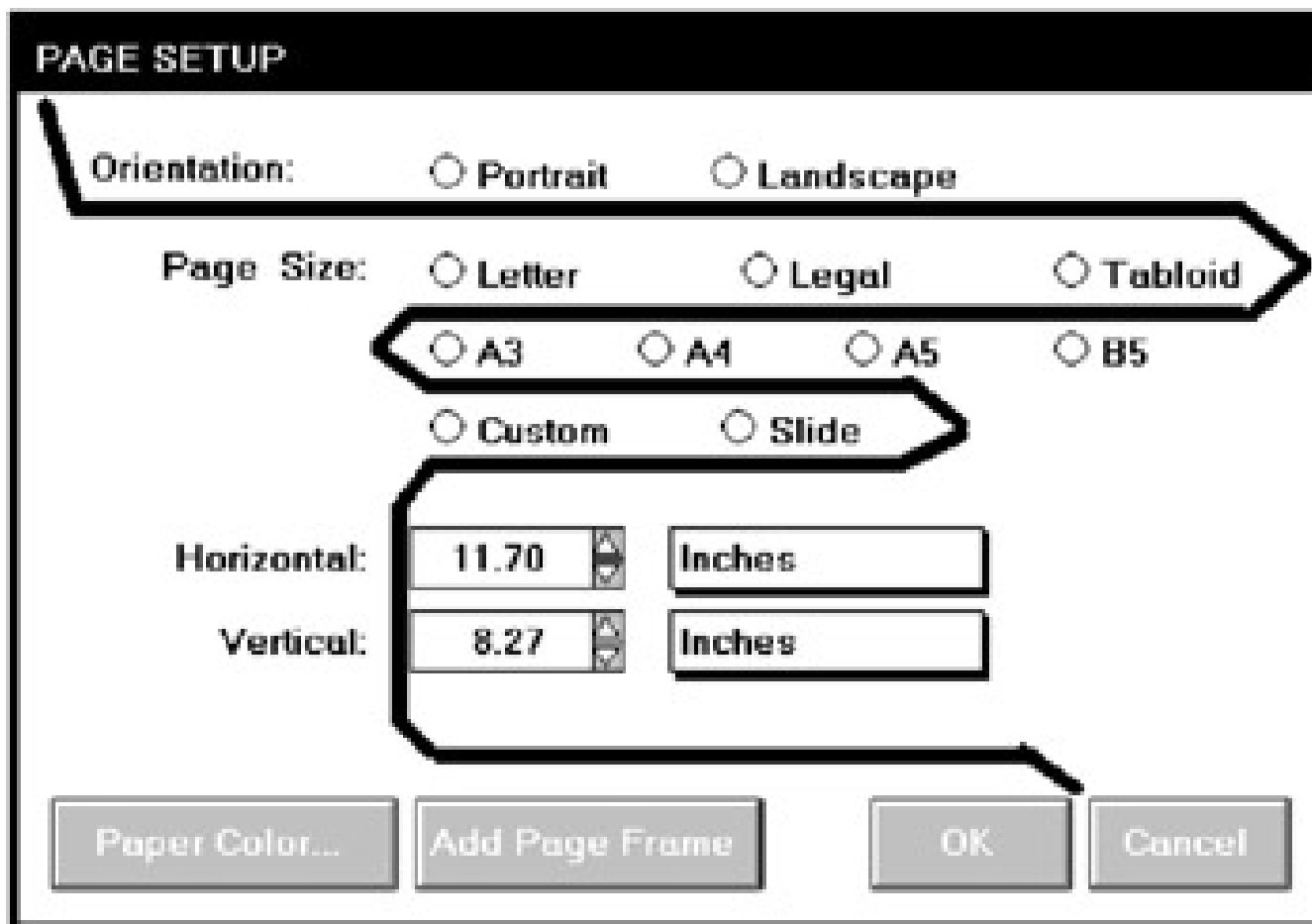
Orientation: Portrait Landscape

Page Size: Letter Legal Tabloid
 A3 A4 A5 B5
 Custom Slide

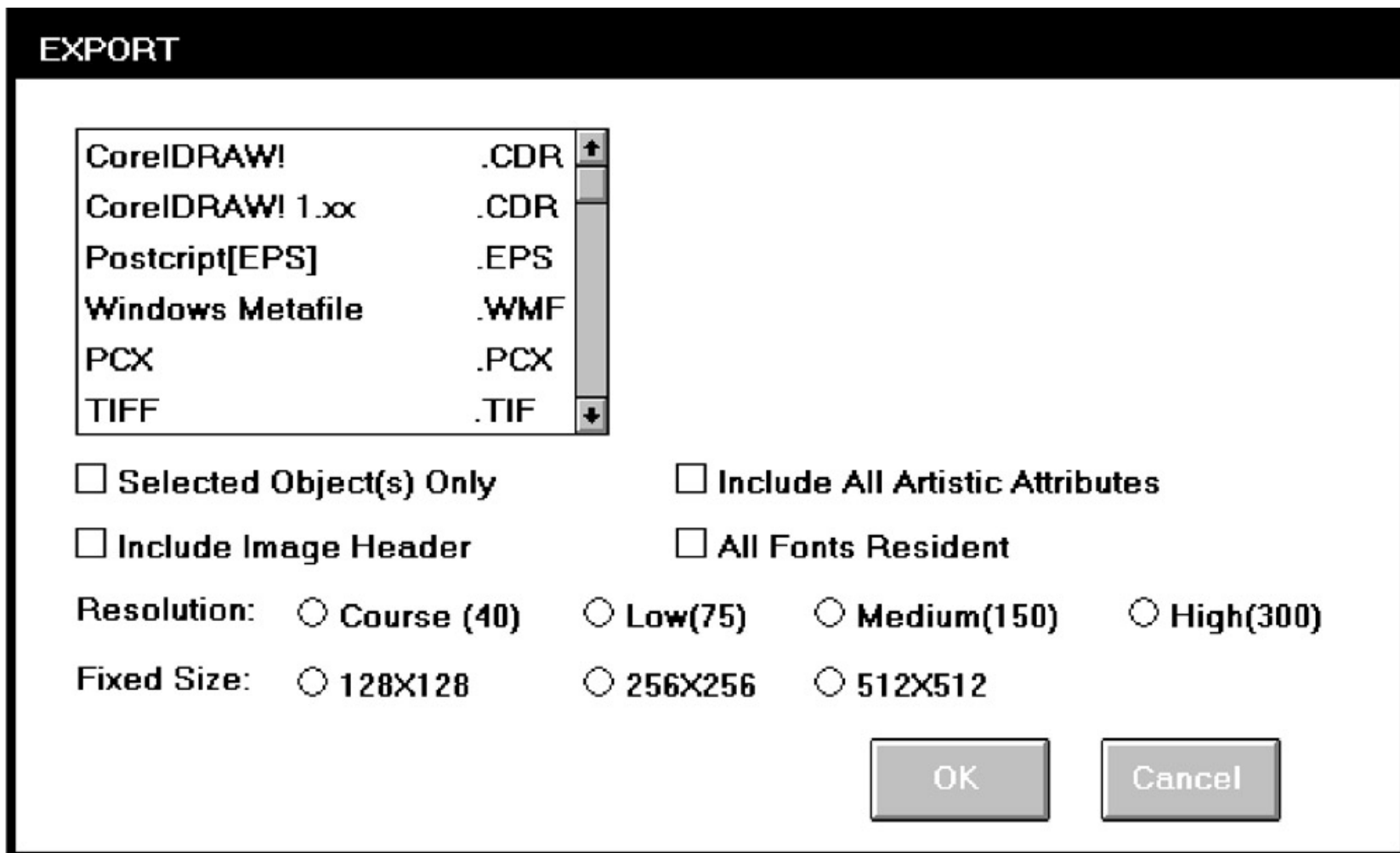
Horizontal: 

Vertical: 

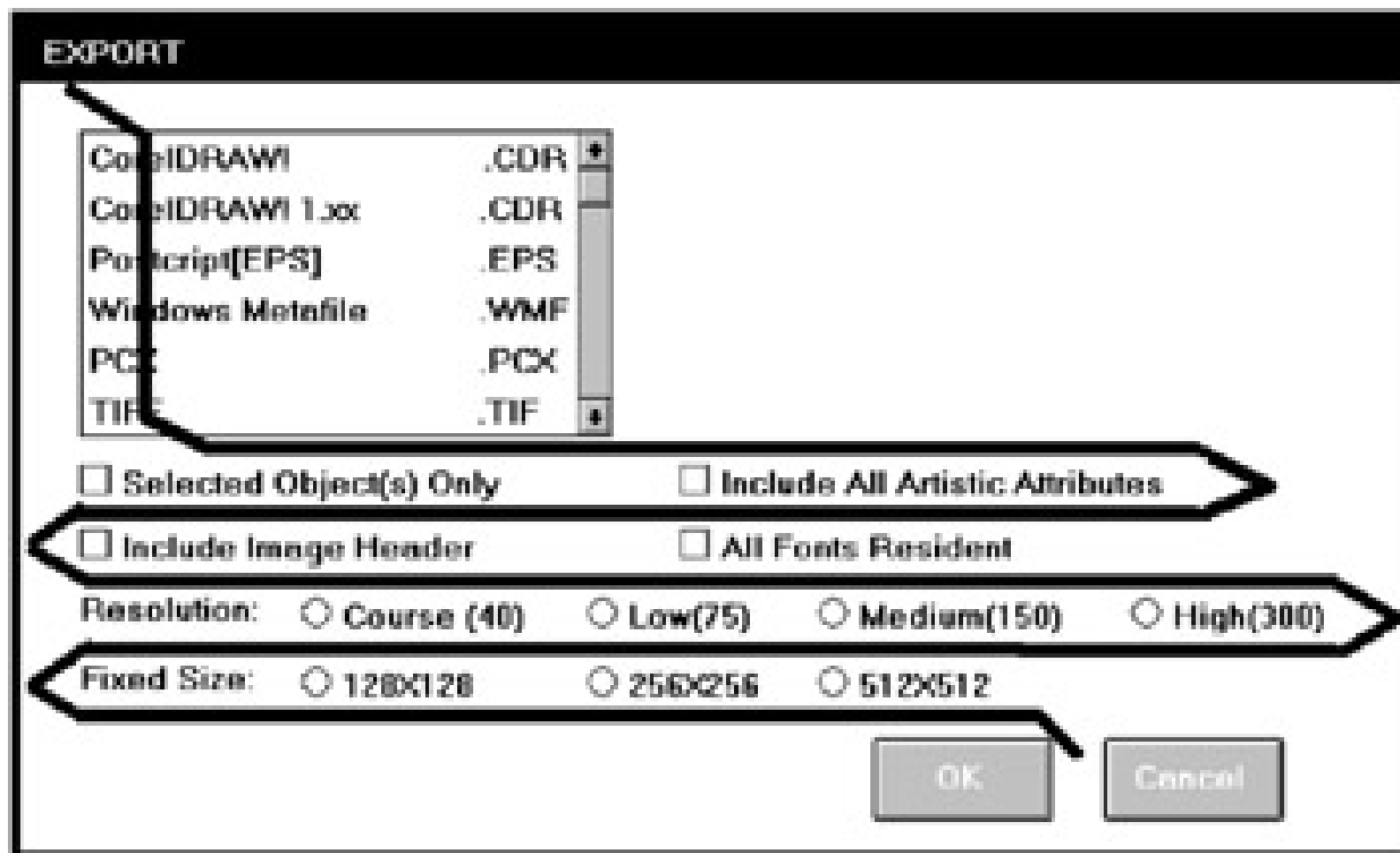
Một số ví dụ về màn hình



Một số ví dụ về màn hình




Một số ví dụ về màn hình





Một số ví dụ về màn hình

Footnote Options

Numbering

Numbering Method:  Characters:

Style in Text: 

Style in Note: 

Restart Numbering on Each Page

Spacing

Line Spacing in Notes:


Spacing Between Notes:

Continued Notes


Print (Continued...) Message

Minimum Note Ht:

Position



Separator



Một số ví dụ về màn hình

The image shows a 'Footnote Options' dialog box with several sections and controls:

- Numbering**
 - Numbering Method: [Dropdown]
 - Character: [Text Field]
 - Style in Text: [Dropdown]
 - Style in Note: [Dropdown]
 - Restart Numbering on Each Page
- Spacing**
 - Line Spacing in Note: [Text Field]
 - Spacing Between Notes: [Text Field]
- Continued Notes**
 - Print (Continued...) Message
 - Minimum Note: [Text Field]
- Position**: [Dropdown]
- Separator**: [Dropdown]
- Buttons: OK, Cancel

Thick black lines are drawn over the dialog, pointing to specific elements: the 'Numbering Method' dropdown, the 'Character' text field, the 'Style in Text' dropdown, the 'Style in Note' dropdown, the 'Restart Numbering on Each Page' checkbox, the 'Line Spacing in Note' text field, the 'Spacing Between Notes' text field, the 'Print (Continued...) Message' checkbox, the 'Minimum Note' text field, the 'Position' dropdown, and the 'Separator' dropdown.



4. Thiết kế màn hình cho Online Pizza Ordering System: đăng nhập, đăng ký người dùng


(a)

(b)




Thiết kế màn hình cho Online Pizza Ordering System: chọn món chính

(c)



Thiết kế màn hình cho Online Pizza Ordering System: chọn món phụ

(d)



Thiết kế màn hình cho Online Pizza Ordering System: xác nhận đơn hàng

(e)



Liệt kê các tình huống tương tác thông qua giao diện của Online Pizza Ordering System

- Using the scenarios in the previous section, the following shows how the new interface handles various situations.
-
-
-