

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
KHOA CÔNG NGHỆ THÔNG TIN

LƯỢNG MẠNH BÁ



TƯƠNG TÁC  
**Người - Máy**  
HUMAN - COMPUTER INTERACTION



NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
KHOA CÔNG NGHỆ THÔNG TIN**

---

**LƯƠNG MẠNH BÁ**

**TƯƠNG TÁC NGƯỜI - MÁY**  
**(HUMAN - COMPUTER INTERACTION)**



**NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT  
HÀ NỘI**

## *Lời mở đầu*

### **<Lần xuất bản thứ nhất>**

Tương tác người - máy (Human Computer Interaction - HCI) nói đơn giản là việc nghiên cứu con người (người dùng), công nghệ máy tính và tác động qua lại giữa các đối tượng đó. HCI là một lĩnh vực được quan tâm từ rất lâu, từ khi máy tính ra đời. Tuy nhiên, do những hạn chế về công nghệ và cách khai thác máy tính nên khoảng từ những năm 50 đến những năm 80 của thế kỷ 20, HCI chưa được quan tâm đúng mức. Thực chất, những người sử dụng lúc đó là các kỹ sư máy tính và cách khai thác chương trình cũng theo kiểu lô (batch): chương trình được gửi khai thác thông qua các thao tác viên của các trung tâm máy tính và hầu như người dùng - các kỹ sư không có đối thoại với chương trình trong thời gian thực hiện chương trình đó. Sau khi có kết quả xử lý, họ mang về phân tích đánh giá theo cách riêng của mình.

Từ những năm 80, khoa học và công nghệ phát triển cùng với sự ra đời của máy vi tính, cách thức làm việc và tương tác đã khác. Máy vi tính ngày càng có nhiều khả năng mạnh hơn: bộ nhớ với dung lượng lớn hơn, tốc độ xử lý nhanh hơn và các thiết bị ngoại vi cũng đa dạng hơn. Nhiều phần mềm ứng dụng với các chức năng phong phú đã xuất hiện đáp ứng nhu cầu ngày càng đa dạng của người dùng và người dùng cũng đa dạng hơn: các kỹ sư máy tính, người dùng đầu cuối,... Đặc biệt là sự xuất hiện của Internet - mạng máy tính toàn cầu đã làm bùng nổ sự khai phá thông tin trên qui mô toàn cầu với nhiều dịch vụ đa dạng và quen thuộc với hầu hết mọi người không chỉ riêng trong giới công nghệ thông tin.

Từ những năm 90, khuynh hướng và nhu cầu đào tạo, học gia tăng. Những nghiên cứu về đào tạo từ xa (E-learning) trên nền Internet phát triển khá mạnh. Các dự án đào tạo từ xa liên quốc gia: Anh-Đức, Anh-Mỹ, Mỹ và của các quốc gia khác đã đưa ra nhiều ý tưởng và các phương pháp đào tạo. Từ những dự án này đòi hỏi các phần mềm phải thân thiện, mang tính tiện dụng cao (usability systems) cho mọi lớp người dùng. Các dự án này đòi hỏi có cách nhìn mới về tương tác người - máy. Các nghiên cứu về lĩnh vực này được

## Lời mở đầu

tăng cường hơn và dành được sự quan tâm của nhiều giới nghiên cứu tâm lý học, khoa học nhận thức của con người và khoa học máy tính.

Từ những năm 90, hiệp hội công nghệ phần mềm về HCI đã tổ chức hội thảo hàng năm, bắt đầu từ SIGCHI 90 (Special Interest Group on Computer-Human Interaction). SIGCHI 1992, New York đã đề xuất khung đào tạo cho HCI trong các trường đại học có ngành Khoa học máy tính [10] và đưa ra dự thảo 4 đề cương môn học cần đào tạo cho HCI.

Vậy tại sao phải nghiên cứu HCI, HCI là gì và liên quan đến những lĩnh vực nào? Máy tính, con người, môi trường, xã hội,...

Hãy thử tưởng tượng một người dùng đầu cuối sử dụng một phần mềm nào đó, do hạn chế về tri thức của mình hay do nhà thiết kế tồi, mà lẽ ra thay vì kích hoạt một chức năng anh ta lại nhấn nhầm và gây ra hậu quả nghiêm trọng. Tại sao máy tính vẫn được coi là *thân thiện, dễ dùng* mà vẫn xảy ra những điều đáng tiếc như vậy. Nếu điều đó thường xảy ra, liệu có ai dám mạo hiểm sử dụng phần mềm không khi mà việc dùng nó quá mệt mỏi và căng thẳng. Vì vậy, máy tính và các thiết bị có liên quan phải được thiết kế với một sự hiểu biết sâu sắc về cái mà con người khi có một ý định trong đầu định sử dụng máy tính và các thiết bị này theo cách thức như chúng là một mạch liên tục với công việc hàng ngày của họ. Để làm điều này, các nhà thiết kế cần phải hiểu rõ những suy nghĩ của con người như họ đang thực hiện các nhiệm vụ theo nghĩa truyền thống và cách thức chuyển các tri thức sẵn có sang một hệ thống thực hiện.

Thuật ngữ tương tác người-máy tuy mới chỉ phổ biến khoảng gần hai thập kỷ gần đây, song nó có nguồn gốc trong nhiều lĩnh vực: công thái học, các yếu tố con người. Các nghiên cứu này có nguồn gốc từ tương tác giữa máy móc và con người (man-machine interaction), sau sang tương tác người - máy tính (human - computer interaction) với sự quan tâm đặc biệt cho máy tính và cộng đồng người dùng. Một lĩnh vực khác của các nghiên cứu do ảnh hưởng của HCI đó là khoa học thông tin và công nghệ. HCI đã thúc đẩy nhiều lĩnh vực khoa học như chúng ta thấy, tuy nhiên nó như là một khái niệm trung tâm trong khoa học máy tính và thiết kế hệ thống.

HCI, không nghi ngờ, nó là một lĩnh vực đa ngành. Người thiết kế một hệ thống tương tác phải có kiến thức đa ngành: tâm lý học, khoa học nhận thức để hiểu được sự cảm nhận thông tin, quá trình nhận thức, kỹ năng giải quyết vấn đề; công thái học để hiểu được

## Lời mở đầu

khả năng vật lý của con người; khoa học máy tính và công nghệ để có thể xây dựng các công nghệ cần thiết; kỹ năng đồ họa để thiết kế các giao tiếp một cách hiệu quả,...

Ở Việt Nam, đây thực sự là một lĩnh vực khá mới mẻ, không nói là hơi xa lạ. Từ năm 1997, Khoa CNTT đã có ý định đưa môn học về HCI vào chương trình đào tạo kỹ sư CNTT của Khoa, song đến năm học 2001-2002 mới bắt đầu thí điểm.

Tuy chậm so với thế giới, song Khoa cũng đã đầu tư và dành sự quan tâm thích đáng cho vấn đề này. Đề tài cấp Nhà nước KC01-09 về xây dựng chương trình đào tạo từ xa qua mạng Internet nhằm xây dựng các bài giảng điện tử cho các môn học trong lĩnh vực CNTT với điểm nhấn là HCI đã bắt đầu triển khai từ năm 2001. Đề tài nhánh tập trung vào 3 chủ điểm có liên quan đến HCI là: HCI, kỹ thuật đồ họa, hiện thực ảo và agent.

Nhằm cung cấp cho sinh viên ngành CNTT các kiến thức và tư liệu tham khảo hữu ích về HCI, theo chỉ đạo của Ban Chủ nhiệm đề tài KC01-09 do GS. TS. Nguyễn Thúc Hải làm chủ nhiệm, tham khảo các nghiên cứu về lĩnh vực này trên thế giới, đặc biệt là sự chỉ đạo của hiệp hội CNPM (Công nghệ phần mềm) về khung tri thức cho HCI như bản đề cương SIGCHI'92 [10] đã nêu, với đặc thù đào tạo và tương lai phát triển của ngành CNTT Việt Nam, chúng tôi cho ra mắt phiên bản đầu tiên của giáo trình tương tác người - máy.

Giáo trình này tập trung vào hai đối tượng chính trong giao tiếp là người dùng con người và máy tính. Với người dùng được hiểu như bộ xử lý thông tin tinh vi: từ cách cảm nhận thông qua các giác quan, cách biểu diễn thông tin đến các hành vi ứng xử để hoàn thành nhiệm vụ. Với máy tính, đối tượng giao tiếp chủ yếu là các thiết bị ngoại vi: bàn phím, các thiết bị trỏ, màn hình và màu sắc. Trên cơ sở tìm hiểu các kỹ thuật giao tiếp, giới thiệu các phương pháp thiết kế, các kỹ thuật đánh giá thông qua các chuẩn về thiết kế, đánh giá nhằm cung cấp kiến thức và các phương pháp luận cho việc xây dựng các hệ thống tương tác có tính tiện dụng, dễ dùng và dễ học theo ISO-9241-11 và ISO 13407.

Với mục đích trên, giáo trình được thiết kế gồm 4 phần và 2 phụ lục. Phần I giới thiệu các khái niệm cơ sở của HCI, gồm 4 chương. Chương 0 là chương mở đầu. Mục đích của chương này là cung cấp các định nghĩa về HCI, các đối tượng tham gia giao tiếp và các vấn đề có liên quan. Đối tượng quan trọng thứ nhất là con người được trình bày trong chương I. Máy tính mà chủ yếu là các thiết bị tham gia giao tiếp được đề cập trong chương II. Chương III dành sự quan tâm cho các kỹ thuật giao tiếp: từ các kỹ thuật truyền thống như

## Lời mở đầu

---

giao tiếp dòng lệnh, menu văn bản đến các kỹ thuật hiện đại, giao tiếp đồ họa GUI (Graphical User Interface), tập trung vào kiểu giao tiếp trực tiếp WIMP (Window, Icon, menu and Pointer).

Phần II tập trung trình bày về các chuẩn trong thiết kế, các kỹ thuật phân tích nhiệm vụ và các mô hình, bao gồm 4 chương. Chương IV giới thiệu về các chuẩn và các mô thức dùng trong thiết kế và các phương pháp thiết kế. Chương V đề cập một số mô hình người dùng: GOMS, KEYSTROKE,... Chương VI giới thiệu về các kỹ thuật phân tích nhiệm vụ, phần quan trọng để hiểu và biểu diễn suy nghĩ và hành động người dùng. Chương VII mô tả các kỹ pháp để biểu diễn đối thoại và ứng dụng.

Phần III trình bày về các kỹ thuật cài đặt và đánh giá tương tác người-máy gồm 2 chương. Vì cài đặt HCI gắn với cài đặt phần mềm, nên trọng tâm của chương VIII là trình bày các kỹ thuật, công cụ và các thư viện quản trị giao tiếp người dùng. Chương IX giới thiệu các kỹ thuật đánh giá giao tiếp người dùng sử dụng trong quá trình thiết kế cũng như đánh giá sản phẩm.

Phần IV là phần nâng cao. Mục đích cung cấp các kiến thức về xây dựng các trợ giúp và nghiên cứu các kỹ thuật trong giao tiếp WEB. Phần này gồm 2 chương: chương X và chương XI. Chương X cung cấp các tiêu chí và cách đánh giá giao tiếp WEB, một ứng dụng phổ biến và người dùng lại rất đa dạng. Chương XI giới thiệu về kỹ thuật thiết kế trợ giúp.

Phụ lục thứ nhất giới thiệu về các công cụ tạo lập mẫu thử dùng trong thiết kế và đánh giá: HyperCard và Revolution. Phụ lục thứ hai minh họa cách thiết kế trợ giúp trong các phần mềm, kèm theo thí dụ minh họa.

Trong quá trình biên soạn, chúng tôi đã nhận được các ý kiến rất quý báu, sự giúp đỡ nhiệt tình của Ban Chủ nhiệm Khoa, các thầy cô trong Khoa, Ban chủ nhiệm đề tài cấp Nhà nước KC01-09, đặc biệt là GS. TS. Nguyễn Thúc Hải, nguyên Trường Khoa - Chủ nhiệm đề tài.

Chúng tôi xin cảm ơn PGS. TS. Nguyễn Ngọc Bình, nguyên Trường Bộ môn Công nghệ phần mềm, Giám đốc Trung tâm Mạng và Thư viện Trường Đại học Bách khoa Hà Nội đã quan tâm và tạo các điều kiện cần thiết cho quá trình biên soạn tài liệu.

## Lời mở đầu

Chúng tôi cũng bày tỏ lòng biết ơn đối với TS. Huỳnh Quyết Thắng - Trưởng Bộ môn Công nghệ phần mềm, phụ trách đề tài nhánh của đề tài KC01-09, ThS. Lê Tân Hùng đã góp nhiều ý kiến và dành nhiều thời gian thảo luận về nội dung cũng như cách trình bày tài liệu.

Chúng tôi cũng xin cảm ơn sự động viên khích lệ của bạn bè, đồng nghiệp trong và ngoài Khoa, nguồn động viên lớn cho chúng tôi hoàn thành quá trình biên soạn đề tài liệu này sớm ra mắt bạn đọc.

Hy vọng tài liệu này sẽ hữu ích cho các bạn sinh viên ngành công nghệ thông tin của các trường đại học trong nước không chỉ trong quá trình học tập trên ghế nhà trường đại học mà còn trong công tác sau này. Chúng tôi cũng hy vọng tài liệu sẽ bổ ích cho các kỹ sư phần mềm đang làm việc trong các doanh nghiệp, các công ty phần mềm.

Để cho giáo trình được hoàn thiện hơn, chúng tôi rất mong nhận được các ý kiến đóng góp của các em sinh viên, các đồng nghiệp xa gần ở các trường đại học cũng như trong các công ty.

Địa chỉ liên lạc: Email: [balm@it-hut.edu.vn](mailto:balm@it-hut.edu.vn)

*Tác giả*

## MỤC LỤC

Lời mở đầu .....	3
Chương 0. Giới thiệu .....	13
0.1. Định nghĩa và khái niệm .....	14
0.2. Người dùng .....	15
0.3. Môi trường làm việc .....	16
0.4. Lĩnh vực liên quan .....	17
0.5. Chất lượng phần mềm .....	19
0.6. Phương tiện và công cụ .....	19
Chương 1. Tâm lý nhận thức của Con người trong giao tiếp và xử lý .....	22
1.1. Tông quan .....	23
1.2. Kênh vào ra .....	23
1.3. Bộ nhớ .....	29
1.4. Lập luận và giải quyết vấn đề .....	33
1.5. Tâm lý và thiết kế hệ thống tương tác .....	35
Chương 2. Nhân tố Máy tính trong tương tác người máy .....	36
2.1. Tông quan .....	37
2.2. Thiết bị vào .....	38
2.3. Thiết bị ra .....	41
2.4. Bộ nhớ .....	44
2.5. Xử lý .....	45
2.6. Tóm lược .....	45
Chương 3. Các mô hình và các dạng tương tác người máy .....	47
3.1. Tông quan .....	48
3.2. Mô hình tương tác .....	48
3.3. Các dạng tương tác .....	50
3.3.1. Giao tiếp dòng lệnh .....	50
3.3.2. Giao tiếp kiểu bảng chọn .....	51
3.3.3. Giao tiếp bằng ngôn ngữ tự nhiên .....	52
3.3.4. Giao tiếp dạng hỏi đáp và truy vấn .....	52
3.3.5. Giao tiếp điện theo mẫu .....	53
3.4. Tương tác WIMP .....	54
3.4.1. Windows .....	55
3.4.2. Biểu tượng (Icons) .....	56
3.4.3. Bảng chọn (Menus) .....	57
3.4.4. Con trỏ (Pointers) .....	58
3.5. Điều khiển trực tiếp .....	58
3.6. Ngữ cảnh tương tác .....	59

## Mục lục

---

<b>Chương 4. Thiết kế giao tiếp Người dùng – Máy tính.....</b>	60
4.1. Tổng quan .....	61
4.2. Các tiếp cận .....	61
4.3. Các vấn đề (paradigms) cho tính dùng được .....	68
4.4. Thiết kế giao tiếp người dùng – máy tính.....	74
4.5. Thiết kế lặp và mẫu thử .....	77
<b>Chương 5. Các mô hình nhận thức người dùng: GOMS &amp; KEYSTROKE.....</b>	84
5.1. Tổng quan .....	85
5.2. Mô hình đặc tả yêu cầu ND.....	85
5.2.1. Mô hình kỹ thuật xã hội (OSTA)/ mô hình thiết kế cộng tác .....	85
5.2.2. Mô hình hệ thống phần mềm (Eason, 1992).....	87
5.2.3. Mô hình đa cách nhìn (Multiview).....	87
5.3. Mô hình nhận thức .....	88
5.3.1 Mô hình GOMS: Goals-Operator-Methods and Selection.....	89
5.3.2. Mô hình ngôn ngữ.....	92
5.3.3. Mô hình Keystroke - KLM .....	93
<b>Chương 6. Đặc tả yêu cầu ND và Phân tích nhiệm vụ .....</b>	99
6.1. Đặc tả nhu cầu người dùng.....	100
6.1.1. Đặc tả chức năng .....	100
6.1.2. Đặc tả dữ liệu.....	101
6.1.3. Đặc tả tính dùng được .....	101
6.2. Công cụ .....	101
6.3. Phân tích nhiệm vụ (Task Analysis) .....	102
6.3.1. Vai trò .....	102
6.3.2. Một số thuật ngữ: Mục đích (Goal), nhiệm vụ (Task) và hành động (Action) .....	102
6.3.3. Kỹ thuật phân tích nhiệm vụ phân cấp (HTA Hierachical Task Analysis)....	103
6.3.4. Kỹ thuật phân tích nhiệm vụ theo nhận thức .....	105
6.3.5. Phân tích nhiệm vụ theo mô hình tri thức - mô hình How to do .....	105
6.4. Thi dụ .....	106
<b>Chương 7. Ký pháp đối thoại và thiết kế.....</b>	110
7.1. Đối thoại là gì? .....	111
7.2. Ký pháp đồ họa .....	112
7.2.1. Mạng dịch chuyển trạng thái - STN .....	112
7.2.2. Mạng dịch chuyển trạng thái phân cấp - HSTN .....	113
7.2.3. Đối thoại tương tranh và bùng nổ tổ hợp .....	114
7.2.4. Lưu đồ luồng (Flow Chart) .....	115
7.2.5. Lưu đồ Jackson - JSD (Jackson Structured Design) .....	115
7.3. Ký pháp văn bản.....	116
7.3.1. Văn phạm .....	116
7.3.2. Luật sản xuất .....	117
7.3.3. CPS và đại số sự kiện .....	118
7.4. Ngữ nghĩa đối thoại và thí dụ .....	120

## Mục lục

---

<b>Chương 8. Hồ trợ cài đặt .....</b>	122
8.1. Tổng quan .....	123
8.2. Các thành phần của Hệ thống Window .....	123
8.3. Lập trình ứng dụng .....	126
8.3.1. Mô thức vòng lặp đánh giá .....	126
8.3.2. Mô thức thông báo tập trung .....	127
8.4. Sử dụng công cụ (Toolkits) .....	129
8.5. Hệ quản trị tương tác người dùng (UIMS) .....	130
8.5.1. UIMS như một kiến trúc khái niệm .....	130
8.5.2. Các xem xét cài đặt .....	131
 <b>Chương 9. Các kỹ thuật đánh giá .....</b>	133
9.1. Khái niệm và vai trò của đánh giá trong thiết kế HCI .....	134
9.2. Các kiểu đánh giá .....	134
9.2.1. Đánh giá trong phòng thí nghiệm .....	134
9.2.2. Đánh giá tại chỗ .....	135
9.3. Đánh giá thiết kế .....	135
9.3.1. Mô hình Cognitive Walkthrough .....	135
9.3.2. Đánh giá Heuristic .....	139
9.3.3. Đánh giá dựa vào xem xét lại quá trình (Review based) .....	141
9.3.4. Đánh giá dựa vào mô hình .....	141
9.4. Đánh giá cài đặt .....	141
9.4.1. Đánh giá thực nghiệm .....	142
9.4.2. Kỹ thuật quan sát .....	144
9.4.3. Kỹ thuật hỏi đáp .....	146
9.5. Lựa chọn phương pháp .....	147
 <b>Chương 10. Multimedia và Web .....</b>	151
10.1. Tổng quan về multimedia và Web .....	152
10.1.1. Multimedia .....	152
10.1.2. Web .....	154
10.2. Các đặc điểm của Web .....	155
10.2.1. Người dùng .....	155
10.2.2. Thông tin và kiến trúc trang Web .....	155
10.3. Các tiêu chí thiết kế và đánh giá trang Web .....	156
10.3.1. Mười lỗi phổ biến .....	156
10.3.2. Các tiêu chí thiết kế và đánh giá trang Web .....	157
10.4. Thiết kế Web .....	164
10.4.1. Thiết kế hướng tới người sử dụng .....	164
10.4.2. Các giúp đỡ định hướng rõ ràng .....	164
10.4.3. Các quy tắc liên kết và điều khiển .....	164
10.4.4. Tạo ngữ cảnh hoặc mắt đọc già .....	165
10.4.5. Dài thông và ảnh hưởng .....	165
10.4.6. Đơn giản và nhất quán .....	166

## Mục lục

10.4.7. Tính ổn định thiết kế .....	166
10.4.8. Phản hồi và đối thoại .....	167
10.4.9. Thiết kế cho các trình duyệt khác .....	167
10.4.10. Chú ý cuối cùng .....	167
10.5. Một số xu hướng phát triển của tương tác người máy trong WEB .....	167
10.5.1. Tính độc lập thiết bị (Device Independence) .....	167
10.5.2. Tương tác đa phương thức (Multimodal Interaction) .....	168
10.5.3. Đồng bộ đa phương tiện (Synchronized Multimedia) .....	168
10.5.4. Trình duyệt bằng giọng nói (Voice Browser) .....	168
10.5.5. Người mì qua mạng .....	169
10.6. Kết luận .....	169
 <b>Chương 11. Thiết kế trợ giúp .....</b>	 170
11.1. Tổng quan về trợ giúp .....	171
11.2. Các yêu cầu về hỗ trợ người dùng .....	172
11.2.1. Tính sẵn dùng .....	172
11.2.2. Tính chính xác và đầy đủ .....	172
11.2.3. Tính nhất quán .....	172
11.2.4. Tính vững chắc .....	172
11.2.5. Tính mềm dẻo .....	173
11.2.6. Tính không lạc hậu? .....	173
11.3. Các tiếp cận để hỗ trợ người dùng .....	173
11.3.1. Lệnh trợ giúp .....	173
11.3.2. Dấu nhắc lệnh .....	174
11.3.3. Trợ giúp ngữ cảnh .....	175
11.3.4. Huấn luyện trực tuyến .....	176
11.3.5. Tài liệu trực tuyến .....	178
11.4. Hệ thống trợ giúp thích nghi .....	182
11.4.1. Biểu diễn tri thức: mô hình người dùng .....	183
11.4.2. Biểu diễn tri thức: mô hình hoá nhiệm vụ và lĩnh vực .....	185
11.4.3. Biểu diễn tri thức: mô hình hoá chiến lược tư vấn .....	186
11.4.4. CÁC KỸ THUẬT BIỂU DIỄN TRI THỨC .....	186
11.5. Thiết kế hệ thống trợ giúp người dùng .....	188
11.5.1. Mô thức trình diễn .....	188
11.5.2. Mô thức cài đặt .....	190
11.6. Ví dụ thiết kế một hệ thống trợ giúp dạng HTML .....	190
Phụ lục A .....	198
Phụ lục B .....	217
Phụ lục C .....	239
Tài liệu tham khảo .....	268

# 0

## GIỚI THIỆU

### Nội dung

1. Định nghĩa và khái niệm
2. Người dùng
3. Môi trường
4. Lĩnh vực liên quan
5. Chất lượng phần mềm
6. Phương tiện và công cụ

## *Chương 0: Giới thiệu*

Khoa học về tương tác giữa con người và máy móc (human-machine interaction) đã được nghiên cứu và phát triển từ lâu nhằm tăng tính tiện dụng của các loại máy móc, công cụ sản xuất, đưa năng suất làm việc lên cao. Những nghiên cứu về tương tác người-máy tính (human-computer interaction) là sự phát triển tiếp theo của khoa học trên trong thời đại hiện nay, khi vai trò của máy tính và các ứng dụng công nghệ thông tin ngày càng trở nên phổ biến. Ở đây ta không quan tâm nhiều tới cấu tạo vật lý của máy tính mà quan tâm tới người dùng con người và các thao tác của họ với máy tính, rút ra các nguyên tắc, các quy luật để có thể phát triển các chương trình ngày càng tiện dụng hơn, đáp ứng tối đa mong muốn của người dùng. Nghiên cứu về tương tác người-máy tính không đơn thuần là nghiên cứu về cách xây dựng giao diện thân thiện với người dùng mà là khoa học để xây dựng, bồi dưỡng chương trình có thể giúp người dùng hoàn thành công việc của họ một cách nhẹ nhàng nhất phụ thuộc vào khả năng vật lý của con người và cải tiến công nghệ.

Kết quả của việc nghiên cứu tương tác người- máy tính giúp ích cho nhà phát triển trong suốt vòng đời của phần mềm: lấy yêu cầu, phân tích, thiết kế, kiểm thử... Những hiểu biết đó giúp cho phần mềm làm ra thỏa mãn hơn các yêu cầu của người dùng đặc biệt là về tính tiện dụng.

Các mô hình lý thuyết mô hình hoá cơ chế tương tác của một cá thể với máy tính. Đó là sự áp dụng lý thuyết để giải thích, xác định những phản ứng của người dùng. Nó không giải quyết các vấn đề về năng lực quan niệm nhưng giúp đỡ người lập trình hiểu cấu trúc nhận thức của con người, xác định nhu cầu của họ, phân tích các đặc tính của một miêu tả hình thức và điều khiển mọi khía cạnh của nhận thức.

Những mô hình ứng dụng, nằm giữa lý thuyết và cụ thể, được sử dụng để đánh giá trước hiệu năng. Nó không bao gồm câu trả lời về quá trình quan niệm mà cho phép tiến hành so sánh các đánh giá trong các hướng có thể.

Trong các phần dưới đây của chương, chúng ta sẽ xem xét một cách chi tiết khái niệm về HCI, các vấn đề liên quan như: người dùng, môi trường, ngữ cảnh giao tiếp, công cụ cho thiết kế xây dựng giao tiếp.

### **0.1 Định nghĩa và khái niệm**

#### **0.1.1. Định nghĩa**

Không có định nghĩa chính xác về tương tác người-máy, tuy nhiên có thể xem xét hai định nghĩa sau. Định nghĩa thứ nhất do Backer và Buxton nêu ra năm 1987:

*"Tương tác người-máy là tập các quá trình, đối thoại và các hành động, qua đó người dùng con người sử dụng và tương tác với máy tính".*

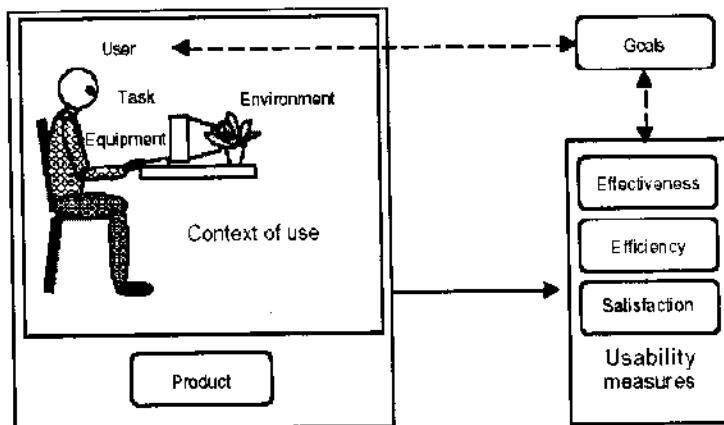
Định nghĩa thứ hai nêu lên cả một quy trình hay vòng đời của tương tác người- máy và cả việc nghiên cứu các hiện tượng xảy ra trên đó. Định nghĩa này do hiệp hội CNPM SIGCHI đưa ra: *"Tương tác người - máy là một lĩnh vực liên quan đến thiết kế, đánh giá và cải đặt hệ thống máy tính tương tác cho con người sử dụng và nghiên cứu các hiện tượng xảy ra trên đó"*.

#### **0.1.2. Vai trò của HCI**

Nhiệm vụ của HCI là tạo ra các hệ thống an toàn và có tính tiện dụng như các hệ

thống chức năng. Tính tiện dụng là một khái niệm trong HCI được hiểu là hệ thống dễ học và dễ dùng. Theo ISO-9241-11, tính tiện dụng là sự mở rộng của một sản phẩm có thể được dùng bởi một số người dùng (ND) chuyên dụng nhằm thực hiện các mục đích chuyên dụng với một ấn tượng (effectiveness), hiệu quả (efficiency) và đáp ứng trong một ngữ cảnh sử dụng riêng biệt.

Tính tiện dụng không phải là một thuộc tính của sản phẩm riêng lẻ mà là một thuộc tính của việc tương tác với sản phẩm trong ngữ cảnh sử dụng. Khung của tính tiện dụng theo chuẩn ISO 9241-11 được minh họa như trong hình 0.1.



Hình 0.1. Khung tính tiện dụng theo ISO 9241-11.

Để dễ vận dụng trong thực tế, Nielson (1993) đưa ra 5 tiêu chí cụ thể cho tính tiện dụng:

- 1) Tính dễ học
- 2) Tính hiệu quả
- 3) Tính dễ nhớ
- 4) Tính dự đoán lỗi
- 5) Đáp ứng tính chủ quan

## 0.2. Người dùng

Trong quá khứ, khi mà máy tính được thiết kế dành cho nghiên cứu khoa học, và những người dùng chủ yếu là các chuyên gia thì việc quan tâm đến giao tiếp là một điều xa xỉ và việc đầu tư cho giao tiếp thường bị bỏ qua. Từ những năm 80 lại đây khi mà máy tính với mục đích đã khác, dần thâm sâu vào mọi lĩnh vực, không chỉ cho nghiên cứu khoa học mà còn nhiều mặt trong đời sống xã hội, trong các công sở, trong lĩnh vực thương mại và trong đào tạo. Nhu cầu trao đổi thông tin ngày càng tăng nhất là khi Internet - siêu xa lộ thông tin ra đời. Người sử dụng máy tính không còn đơn thuần là các chuyên gia mà tất cả mọi người có nhu cầu: từ người làm công tác quản lý, các thương gia đến cả các trẻ em và lan đến các tầng lớp nông dân ở mọi vùng. Do vậy, các nhà thiết kế phần mềm không thể bỏ qua việc thiết kế giao tiếp khi chỉ dành đặc quyền cho chức năng hệ thống. Giao tiếp và

## **Chương 0: Giới thiệu**

chức năng là hai mặt có tính tương hỗ và bổ sung cho nhau. Nếu giao tiếp tồi thì chức năng sẽ mờ mịt, không rõ ràng; ngược lại nếu giao tiếp được thiết kế tốt nó sẽ cho phép các chức năng của hệ thống hỗ trợ người dùng hoàn thành nhiệm vụ, nhất là khi đa số người dùng hiện nay có ít tri thức về máy tính.

Vì vậy, việc hiểu tâm lý nhận thức của người dùng, cách thức mà họ suy nghĩ và thực hiện nhiệm vụ là điều quan tâm trước tiên. Tiếp theo đó là việc thiết kế các thiết bị, nhất là các thiết bị phục vụ cho việc giao tiếp mà quan trọng hơn cả là các thiết bị vào và các thiết bị hiển thị thông tin ra. Những điều này sẽ tạo một môi trường đa dạng, thuận tiện cho người dùng.

### **0.3. Môi trường làm việc**

Từ những năm 50 đến khoảng cuối thập kỷ 70 của thế kỷ 20, các máy tính chủ yếu là các máy tính lớn (Mainframe) hoạt động độc lập. Người sử dụng máy tính không phải là người thao tác điều khiển máy tính. Những người sử dụng thông qua các thiết bị vào nối với máy tính như các thiết bị đọc bìa, băng giấy hay thông qua các thiết bị cuối (terminal) để giao tiếp với máy tính. Theo cách này, hầu như họ không có trao đổi thông tin nào khác ngoài cái đã phải chỉ định rõ từ trước. Và nếu họ muốn thì cũng không thể thực hiện được do hạn chế về công nghệ.

Việc giao tiếp nếu có cũng rất nghèo nàn ngay cả đối với các thao tác viên, người trực tiếp điều khiển máy tính. Người dùng chỉ có thể giao tiếp dưới dạng văn bản, dạng các câu lệnh. Rõ ràng cách thức này là rất hạn chế và làm cho máy tính khó thâm nhập vào các lĩnh vực.

Tuy nhiên, điều này đã hoàn toàn đổi khác: song song với các máy tính lớn, máy mini, máy vi tính đã ra đời vào đầu thập niên 80 và ngày càng hoàn thiện hơn về công nghệ và tính năng. Các máy tính đã được nối mạng: từ các mạng cục bộ, mạng diện rộng đến Internet. Việc nối mạng, nhất là nối Internet đã tạo điều kiện cho việc trao đổi thông tin dễ dàng hơn không chỉ trong một nhóm người trong một cơ quan, một quốc gia mà đã nâng lên quy mô toàn cầu. Điều này đã làm cho máy tính và nhu cầu dùng máy tính trở nên phổ biến hơn.

Các công nghệ hiện đại làm cho môi trường máy tính trở nên đa dạng hơn. Thông tin không đơn thuần là dạng văn bản truyền thống mà còn có các dạng khác với diện mạo phong phú như đồ họa, hình ảnh, âm thanh. Theo cách nói của các nhà tin học là đa phương tiện (multimedia) và siêu văn bản (hypertext). Giao diện người dùng đồ họa GUI đã trở nên gần gũi và quen thuộc với người dùng.

Nhờ các tiến bộ của công nghệ, khả năng thiết kế các phần mềm ngày càng trở nên sinh động với giao tiếp thân thiện là điều có thể làm được. Vẫn đề là còn tuỳ thuộc vào khâu thiết kế giao tiếp và tri thức của các kỹ sư phần mềm. Công nghệ và phương tiện cho phép thiết kế các tương tác với nhiều lựa chọn phong phú, hợp với thị giác và thói quen của người dùng. Song đó mới chỉ là một phần. Xét về góc độ hiệu năng, càng nhiều phần mềm hoạt động cùng một lúc thì tốc độ xử lý càng bị giảm và cần nhiều tài nguyên hơn. Hơn nữa tài

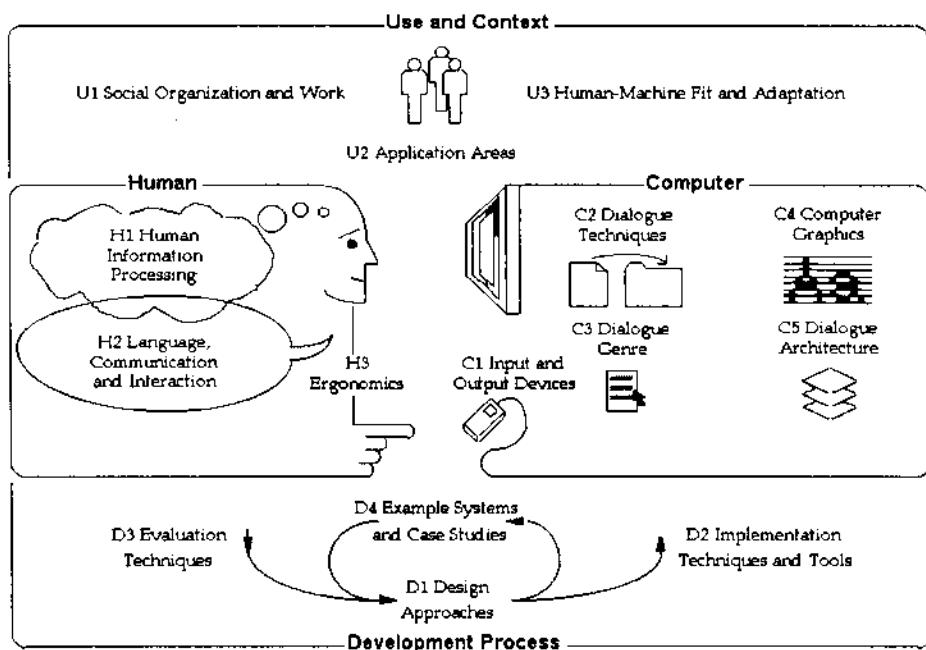
## *Chương 0: Giới thiệu*

nguyên của máy tính cũng có hạn. Do vậy cần thiết kế để sử dụng tài nguyên một cách tối ưu.

### **0.4. Lĩnh vực liên quan**

Trước đây, việc xây dựng các phần mềm máy tính liên quan chủ yếu đến phương pháp tính. Thường các bài toán khoa học kỹ thuật được giải quyết qua việc mô hình hóa bằng các mô hình toán học thích hợp. Tiếp theo là sử dụng các ngôn ngữ lập trình để cài đặt phần mềm.

Ngày nay, các lĩnh vực đa dạng hơn: phương pháp tính, tính toán kỹ hiệu, soạn thảo văn bản, xử lý đồ họa, hình ảnh, âm thanh, đa phương tiện... Có sự đan xen nhau giữa dữ liệu, dòng lệnh và kết quả tạo nên một sự đổi mới phong phú và phức tạp. Để cập đến các lĩnh vực liên quan, hiệp hội Công nghệ phần mềm SIGCHI, 1992 tại New York đã đưa ra bốn lĩnh vực chính: môi trường, con người, máy tính và quá trình phát triển như chỉ ra trong hình 0.2.



**Hình 0.2. Các lĩnh vực liên quan (SIGCHI, 1992 New York).**

#### *0.4.1. Môi trường và ngữ cảnh*

Đó là tập những yếu tố mà máy tính cung cấp, trong đó ứng dụng hoạt động, cung cấp các chức năng và giao tiếp cho người dùng để hoàn thành nhiệm vụ của mình. Các tổ chức xã hội, công việc và kinh doanh tương tác với ứng dụng đó. Các lĩnh vực ứng dụng được phân theo các tiêu chí:

## *Chương 0: Giới thiệu*

---

- Cá nhân hay nhóm
- Giao tiếp hướng văn bản
- Giao tiếp hướng truyền thông
- Trợ giúp trực tuyến hay điều khiển hệ thống liên tục
- Trợ giúp thiết kế CAD/CAM, ...

### *0.4.2. Con người*

Con người là nhân tố chính mà việc thiết kế phần mềm và tương tác phải lấy nó làm trung tâm. Mục đích chính trong nghiên cứu con người là hiểu con người như một bộ xử lý thông tin. Thông qua giao tiếp, con người nhận thức và hiểu cách thức phải hành động để đạt mục đích. Con người có nhiều thói quen tiềm ẩn và nhiều khi phản xạ theo thói quen và các suy nghĩ kiểu tự nhiên của mình.

Một yếu tố khác cũng cần quan tâm đó là vấn đề công thái học. Công thái học bắt nguồn từ lao động ngành nghề, liên quan tới môi trường và điều kiện làm việc. Nó cũng liên quan tới khả năng nhận thức và giới hạn vật lý của con người, thí dụ về khả năng nhớ. Theo những thống kê, con người chỉ có khả năng nhớ được từ 5 đến 9 sự việc liên tiếp, và trung bình là 7. Chi tiết về vấn đề này sẽ được đề cập trong chương Một.

### *0.4.3. Máy tính và kiến trúc tương tác*

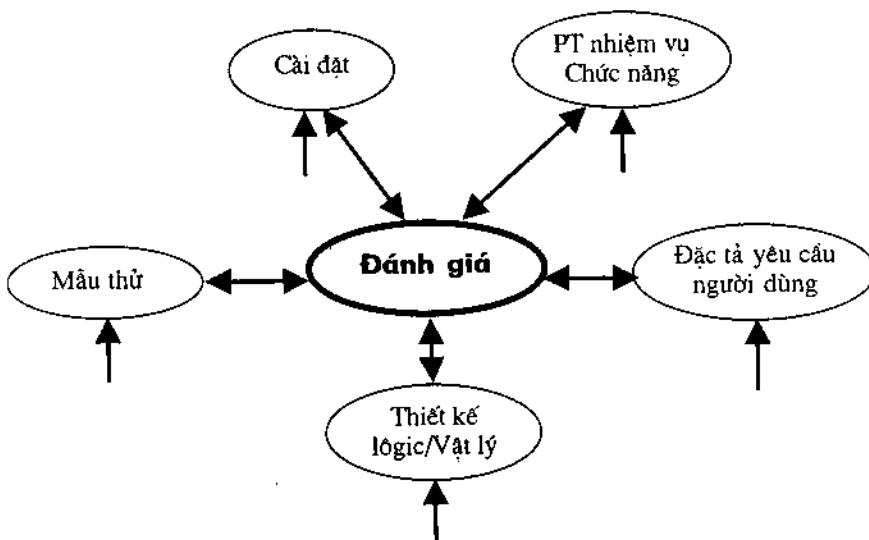
Chúng ta không quan tâm đến kiến trúc vật lý của máy tính mà chỉ tập trung xem xét các thiết bị có liên quan đến tương tác. Đó là các thiết bị vào và hiển thị thông tin. Các tương tác được thực hiện thông qua các đối thoại. Có nhiều kỹ thuật đối thoại đã được nghiên cứu và phát triển: dạng văn bản như ngôn ngữ dòng lệnh, menu đến các giao tiếp đồ họa, giao tiếp kiểu điều khiển trực tiếp. Phần liên quan đến các thiết bị vào ra sẽ được trình bày trong chương Hai; các kỹ thuật tương tác được đề cập trong chương Ba.

Để có thể thiết kế giao tiếp chúng ta cũng cần đến các kỹ thuật cho phép mô tả đối thoại và các ký pháp để biểu diễn đối thoại. Phần này sẽ được trình bày trong hai chương Sáu và Bảy.

### *0.4.4. Quy trình phát triển*

Cũng giống như trong thiết kế phần mềm, việc thiết kế và phát triển tương tác người-máy cũng diễn ra qua nhiều giai đoạn được gọi là vòng đời.

Hình 0.3 biếu diễn vòng đời hình sao trong thiết kế và phát triển tương tác người-máy. Chi tiết của phần này được đề cập trong chương Bốn.



**Hình 0.3. Vòng đồi hình sao của HCI (Hix & Hartson, 1993).**

#### 0.5. Chất lượng phần mềm

Chất lượng phần mềm là một khái niệm cần được định nghĩa rõ ràng và nó thuộc lĩnh vực đánh giá phần mềm. Chúng ta dùng ngữ nghĩa chất lượng ở đây với hàm ý so sánh để hiểu được vai trò của thiết kế giao tiếp người dùng.

Trước đây, chất lượng được hiểu là trạng thái ứng xử đúng của phần mềm khi người dùng cung cấp dữ liệu hay thao tác đúng. Tiếp theo, người ta có yêu cầu cao hơn: trạng thái ứng xử vẫn đúng khi mà dữ liệu hay thao tác có thể sai.

Ngày nay, ngoài những yêu cầu trên, giao tiếp phải thân thiện, sinh động và dễ dùng. Nó phải đáp ứng với nhu cầu đa dạng của mọi lớp người dùng và thời gian đào tạo hay học là ngắn nhất.

Trong tài liệu này, chúng tôi chỉ đề cập đến chất lượng giao tiếp thông qua các kỹ thuật đánh giá và các tiêu chí cho các loại giao tiếp khác nhau. Chi tiết sẽ được đề cập qua chương Bốn: các chuẩn hoá và nguyên tắc thiết kế và chương Tám: các kỹ thuật đánh giá giao tiếp người dùng.

#### 0.6. Phương tiện và công cụ

Trong tương tác, hai thành phần cơ bản tham gia giao tiếp đó là con người, người dùng và máy tính. Phương tiện giao tiếp giữa hai đối tượng đó là đối thoại thông qua môi trường là phần mềm. Cũng như giao tiếp giữa hai hay nhiều người với nhau được thực hiện thông qua tiếng nói, hành động hay chữ viết. Giao tiếp này có thể là trực tiếp như khi hai con người đàm thoại với nhau mặt đối mặt, cũng có thể là gián tiếp nếu khoảng cách xa bằng cách gửi thư. Tuy nhiên giao tiếp trong tương tác người dùng – máy tính có khác.



**Hình 0.4. Đối thoại giữa người dùng và máy tính.**

Người dùng thông qua các phương tiện giao tiếp cung cấp bởi phần mềm, tương tác với máy tính để đưa các yêu cầu (có thể là dữ liệu hay mệnh lệnh) nhằm thực hiện nhiệm vụ của mình. Máy tính sẽ phân tích yêu cầu, căn cứ vào các chức năng đã định sẵn, thực hiện các yêu cầu đó bằng cách trao đổi giữa các module (hình 0.4). Việc trao đổi giữa các module là trong suốt với người dùng. Việc đáp lại hay sự trả lời của hệ thống thông qua việc biến đổi trạng thái của mình thể hiện trên thiết bị hiển thị. Điều quan trọng ở đây là làm sao, qua giao tiếp người dùng hiểu công việc mình làm và biết thực hiện công việc đó bằng cách nào. Họ phải nhận biết được tác động của hệ thống và tuỳ tình huống bằng nhiều cách khác nhau như sự xuất hiện một thông báo, hay một hình ảnh, sự đổi màu hay biến mất của một hình, sự che mờ đi của một thực đơn, . . .

Như vậy, việc xây dựng giao tiếp qua các kỹ thuật đối thoại là công việc không đơn giản. Người kỹ sư phần mềm phải biết lựa chọn công cụ thích hợp. Công cụ ở đây chính là các ngôn ngữ lập trình.

Các ngôn ngữ lập trình hiện nay khá đa dạng và có khả năng đồ họa khá mạnh đáp ứng được các đòi hỏi trên về giao tiếp. Người ta khuyến cáo có thể dùng Visual Basic, Visual C hay Java. Hai ngôn ngữ đầu tiên có nhược điểm là phụ thuộc vào môi trường. Điều này sẽ ảnh hưởng đến việc xây dựng các hệ thống phần mềm. Java là ngôn ngữ lập trình hướng đối tượng ra đời tháng 5/1995 do nhóm chuyên gia đứng đầu là James Gosling của Công ty Sun Microsystem phát minh.

Lúc đầu, Java được thiết kế để xây dựng các ứng dụng trên Internet như WEB. Về sau Java được phát triển để xây dựng các ứng dụng độc lập. Điểm mạnh đặc biệt của Java là độc lập với hệ nền. Nó hoạt động tốt trên mọi hệ điều hành hỗ trợ Java Virtual Machine.

Khả năng đồ họa của Java khá tốt. Các phiên bản Java trước 1.2 cung cấp thành phần đồ họa được xây dựng trong thư viện AWT (Abstract Windows Toolkits). AWT là thư viện hữu ích để xây dựng các chương trình ứng dụng giao tiếp người dùng đồ họa. Tuy nhiên AWT phụ thuộc nhiều vào hệ nền. Do vậy, từ phiên bản 1.3 trở đi, thành phần giao tiếp người dùng được thay thế bởi tập hợp các thành phần linh hoạt, đa năng và mạnh mẽ: thành phần SWING. SWING vẫn kế thừa các tính năng của AWT, song đã cải tiến thêm và bổ sung thêm các lớp mới.

Chương trình đồ họa Java được điều khiển bởi biến cò. Khi chương trình vận hành tức là tương tác với người dùng thì tất mọi biến cò chỉ phôi việc thi hành chương trình. Biến cò phát sinh từ các tác động bên ngoài của con người như: bấm chuột, nhấn phím..., hay từ

## *Chương 0: Giới thiệu*

---

Hệ điều hành. Các lớp biến cố được chứa trong gói Java.awt.event ngoại trừ một số biến cố như : EventListenerList, ListSelectionEvent chứa trong gói javax.Swing.event.

Swing quản lý các đối tượng lắng nghe thông qua ListenerList, nó được gọi từ EventListenerList. Khi nhận được biến cố phát sinh do tác động lên các đối tượng nguồn thì đối tượng lắng nghe (listener) đã được đối tượng nguồn đăng ký trước sẽ nhận và xử lý biến cố dựa theo chương trình đã lập. Giữa đối tượng nguồn và đối tượng lắng nghe luôn liên lạc với nhau thông qua phương thức xử lý biến cố (handler) đặt trên đối tượng lắng nghe chi tiết về các biến cố, cách thức xử lý cũng như chi tiết về Java SWING sẽ được minh họa trong phần phụ lục.

**1**

**TÂM LÝ NHẬN THỨC CỦA  
CON NGƯỜI TRONG GIAO TIẾP  
VÀ XỬ LÝ**

**Nội dung**

- 1.1. Tổng quan**
- 1.2. Kênh vào ra**
- 1.3. Bộ nhớ**
- 1.4. Lập luận và giải quyết vấn đề**
- 1.5. Tâm lý và thiết kế hệ thống tương tác**

## 1.1. Tổng quan

Con người, người dùng là một đối tượng mà các hệ thống máy tính được thiết kế để trợ giúp và đáp ứng các yêu cầu của họ. Về góc độ khoa học nhận thức, con người được xem như một hệ thống xử lý thông tin tinh tế gồm 3 hệ thống con (Card, Moral and Newell, 1983):

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

Hệ thống cảm nhận điều khiển các kích thích tới từ thế giới bên ngoài qua các kênh vào như thị giác, thính giác và cảm giác, ... Hệ thống xử lý quản lý các hành động. Còn hệ thống nhận thức cung cấp các xử lý cần thiết để liên kết hai hệ thống trên. Cách thức tương tác được thực hiện qua 3 giai đoạn: cảm nhận, lưu trữ và xử lý. Chính vì vậy mà mỗi hệ thống con đều có bộ nhớ và bộ xử lý riêng của mình. đương nhiên, mức độ phức tạp là khác nhau, phụ thuộc vào nhiệm vụ mà hệ thống phải thực hiện.

Với mục đích tìm hiểu nhận thức và cách xử lý thông tin của con người nói chung, cũng như các hạn chế vật lý về khả năng của con người, để có thể xây dựng được các hệ thống phần mềm có tính dùng được, trong chương này, chúng ta tập trung xem xét các nội dung chính như: các kênh vào ra qua đó con người cảm nhận thông tin, bộ nhớ nơi sẽ lưu trữ thông tin tạm thời hay nơi diễn ra các xử lý. Hai nội dung này sẽ được giới thiệu trong các phần 1.2 và 1.3. Tiếp theo, chúng ta sẽ xem xét cách thức lập luận và giải quyết vấn đề trong phần 1.4. Cuối cùng, phần 1.5, đề cập đến sự liên quan giữa tâm lý nhận thức với việc thiết kế các hệ thống giao tiếp.

## 1.2. Kênh vào ra

Con người tương tác với thế giới bên ngoài thông tin nhận được và đáp ứng bằng thông tin gửi đi. Trong quá trình tương tác với máy tính, người dùng nhận được thông tin gửi ra bởi máy tính và đáp ứng bằng cách cung cấp thông tin vào cho nó. Như vậy đâu ra của người dùng trở thành đầu vào của máy tính và ngược lại. Để tránh nhầm lẫn, chúng ta dùng thuật ngữ kênh vào-ra để mô tả cho sự tương tác này.

Kênh vào của con người diễn ra chủ yếu nhờ các giác quan. Còn kênh ra nhờ mô tơ điều khiển các tác động thông qua các giác quan vật lý như: tay, mắt, tai, ... Trong 5 giác quan của con người, có 3 giác quan quan trọng với tương tác người-máy tính, đó là: thị giác (nhìn), thính giác (nghe) và xúc giác (sờ, nếm, nhấn phím). Tương tự như vậy các tác nhân có ảnh hưởng đến đầu ra đó là: chân tay, ngón tay, mắt, đầu và hệ thống ngữ âm. Tuy nhiên, trong tương tác với máy tính, ngón tay là chủ yếu nhất. Người dùng sử dụng ngón tay để nhấn phím hay điều khiển chuột.

Thứ hình dung khi làm việc với một máy tính cá nhân với thiết bị vào chính là bàn phím và chuột. Các ứng dụng mà bạn khai thác thường có giao tiếp đồ họa với menu, biểu tượng và cửa sổ. Trong tương tác với ứng dụng, thông tin mà chúng ta nhận được thông qua quan sát là chính. Tất nhiên chúng ta có thể cảm nhận bằng nghe qua các âm thanh mà máy tính phát ra, song điều đó hiện tại còn khá ít. Nhấn phím đóng vai trò quan trọng trong giao tiếp như nhấn phím để đáp lại yêu cầu của hệ thống hay cung cấp dữ liệu. Chính vì vậy,

## **Chương I: Tâm lý nhận thức của con người trong giao tiếp và xử lý**

trong các phần tiếp sau chúng ta sẽ xem xét chi tiết vai trò của một số giác quan quan trọng trong giao tiếp: thị giác, thính giác, xúc giác và chuyển động.

### **1.2.1. Quan sát**

Việc quan sát của con người là một hoạt động phức tạp với các giới hạn về khả năng vật lý và cảm nhận. Quá trình cảm nhận có thể coi như gồm hai giai đoạn: giai đoạn cảm nhận và giai đoạn xử lý, giải nghĩa các kích thích. Giai đoạn cảm nhận nhận các kích thích vật lý từ thế giới bên ngoài và chuyển cho giai đoạn tiếp theo. Các tính chất vật lý của các kích thích mà người cảm nhận được sẽ được phân tích theo kích thước, màu sắc, độ sáng và độ tương phản.

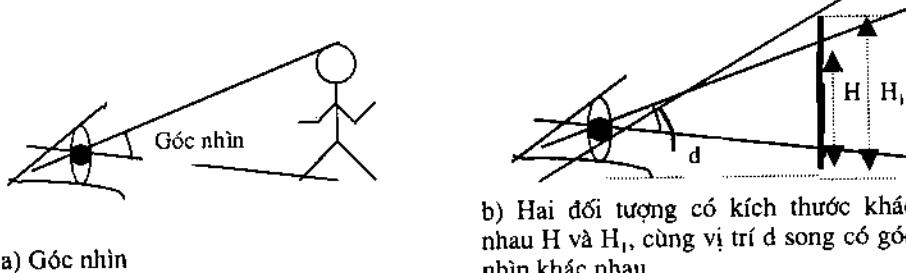
Mắt người là một bộ máy khá tinh vi và phức tạp. Tuy nhiên, chúng ta không đi sâu vào cấu trúc của nó mà chỉ xem việc tạo ảnh trên võng mạc như thế nào. Trên võng mạc có hai loại tế bào: tế bào hình nón và tế bào hình que. Tế bào hình que khá nhạy cảm với ánh sáng và nó cho phép chúng ta nhìn thấy đối tượng dù dưới ánh sáng yếu. Ngược lại, các tế bào hình nón kém nhạy cảm với ánh sáng hơn. Có ba loại tế bào hình nón cảm nhận các tia sáng với bước sóng khác nhau giúp ta cảm nhận được màu sắc. Ba loại đó cảm nhận ba màu cơ bản: màu đỏ với bước sóng dài, màu lục với bước sóng trung và màu tím với bước sóng ngắn.

Vấn đề quan trọng trong tương tác là xem xét sự phụ thuộc của cảm nhận vào kích thước hay khoảng cách tương đối giữa đối tượng quan sát và mắt, vào màu sắc, độ sáng và độ tương phản của đối tượng như thế nào, cũng như các khả năng và hạn chế của hệ thống thị giác của con người.

#### *i) Cảm nhận về kích thước, khoảng cách*

Thông tin mà hệ thống thị giác nhận được phải được lọc và chuyển cho các phần tử xử lý. Các phần tử này cho phép chúng ta nhận biết được cảnh quan một cách rõ ràng, loại bỏ các nhập nhằng và phân biệt màu sắc. Tuy nhiên, sự cảm nhận chính xác phụ thuộc trước tiên vào kích thước đối tượng và khoảng cách từ đối tượng đến mắt.

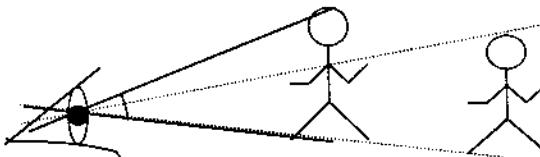
Chúng ta biết rằng ảnh đối tượng trên võng mạc là ảnh ngược và dù khoảng cách xa hay gần nhưng nằm trong giới hạn nhìn thấy, mắt điều tiết và chúng ta nhận rõ được đối tượng. Kích thước của hình ảnh được đặc tả bởi góc nhìn (visual angle). Hình 1.1 dưới đây minh họa cách tính góc nhìn.



**Hình 1.1. Minh họa góc nhìn.**

Góc nhìn là góc giới hạn bởi hai đường thẳng từ đỉnh và từ chân đối tượng đi qua tâm nhìn. Góc nhìn thường được đo bằng độ hay phút ( $1^{\circ}$  bằng  $60'$ ) hay giây ( $1'$  bằng  $60''$ ). Góc nhìn phụ thuộc vào kích thước đối tượng và khoảng cách từ đối tượng đến mắt.

Nếu hai đối tượng ở cùng khoảng cách, đối tượng nào có kích thước lớn hơn, đối tượng đó có góc nhìn tốt hơn, có nghĩa là quan sát tốt hơn (hình 1.1b). Tương tự, hai đối tượng có cùng kích thước, đặt ở khoảng cách khác nhau sẽ có góc nhìn khác nhau: đối tượng ở xa hơn, góc nhìn nhỏ hơn (hình 1.1c).



**Hình 1.1c. Hai đối tượng cùng kích thước, đối tượng ở xa có góc nhìn nhỏ hơn.**

Như vậy, góc nhìn có ảnh hưởng thế nào đến sự cảm nhận của chúng ta về kích thước. Trước hết, nếu góc nhìn quá nhỏ chúng ta sẽ không cảm nhận được đối tượng. Trong quan sát, người ta hay dùng một khái niệm gọi là “độ nhìn”. Độ nhìn là khả năng mà một người bình thường cảm nhận được các chi tiết của đối tượng. Cho rằng góc nhìn của đối tượng sẽ giảm khi đối tượng ở xa (hình 1.1c), song hy vọng chúng ta vẫn có thể cảm nhận được đối tượng như nó chỉ nhỏ hơn thôi.

Tóm lại, sự cảm nhận về kích thước đối tượng là một hằng số, ngay cả khi góc nhìn thay đổi. Như vậy, chiều cao của một con người được cảm nhận như là hằng số ngay cả khi họ chuyển động ra xa hơn. Đó chính là *luật hằng số* của kích thước. Điều này gợi ý cho chúng ta rằng, sự cảm nhận về kích thước liên quan đến các yếu tố khác hơn là góc nhìn.

### *ii) Cảm nhận độ sáng tối*

Độ sáng tối là đáp ứng chủ quan của mức độ sáng. Nó phụ thuộc vào số tia sáng phát ra từ đối tượng, chính xác hơn là phụ thuộc vào số tia sáng rơi trên bề mặt đối tượng và tính chất phản xạ của bề mặt. Độ sáng là một đặc tính vật lý quan trọng và có thể đo bằng quang kế (photometer). Độ tương phản là một khái niệm quan trọng trong thị giác, nó được hiểu là độ nỗi của ánh đối tượng so với nền.

Tuy độ sáng tối là phản ứng khách quan, song nó cũng giúp ta phân biệt sự khác nhau về mức sáng. Hệ thống thị giác của chúng ta có khả năng tự điều chỉnh với các thay đổi về độ sáng tối. Vì có ít tế bào hình que trong hốc mắt nên khi ánh sáng tối, ta khó nhìn đối tượng hơn.

Tóm lại, độ nhìn tăng khi mức sáng tăng. Điều này làm ta cần lưu tâm khi sử dụng các thiết bị hiển thị với mức sáng cao. Tuy nhiên, khi mức sáng tăng thì độ lấp loè cũng tăng.

### *iii) Cảm nhận màu*

Màu sắc là yếu tố thứ ba cần được xem xét trong quan sát. Khi đề cập đến màu, người ta thường chú ý đến ba thành phần:

- Hue: Sắc thái màu
- Intensity: Cường độ màu
- Saturation: Độ bão hòa

Với một nguồn sáng đơn sắc, hue phụ thuộc vào bước sóng của ánh sáng; độ bão hòa sẽ thay đổi nếu ta tăng thêm lượng ánh sáng trắng. Mắt người có thể phân biệt được khoảng 150 màu, song khi thay đổi sắc màu và cường độ sáng, chúng ta lại có thể cảm nhận tới hàng triệu màu [2]. Người ta cũng nhận thấy rằng với một người ít đào tạo, số màu cảm nhận được giảm chỉ khoảng 10 màu.

Cũng cần chú ý hiện tượng mù màu, tức là không có khả năng cảm nhận màu sắc. Theo thống kê có khoảng 8% đàn ông và 1% phụ nữ bị mù màu.

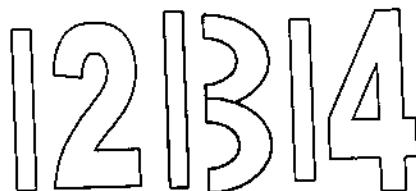
*iv) Khả năng và hạn chế của quá trình cảm nhận*

Ngoài các yếu tố tác động như trên, trong cách thức cảm nhận đối tượng nhiều khi chúng ta còn áp đặt ý chủ quan của mình. Thí dụ, khi đã biết rõ kích thước một đối tượng đặc biệt nào đấy, chúng ta sẽ cảm nhận nó ở kích thước đó dù đối tượng đó ở xa hay cprox; gần.

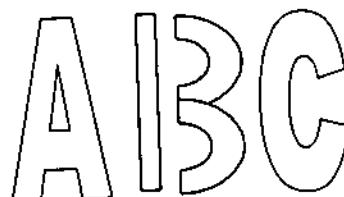
Trong quá trình quan sát đối tượng, lúc ta đứng yên còn vật thể lại chuyển động hoặc vật thể đứng yên mà ta lại chuyển động. Hệ thống xử lý thị giác có khả năng điều chỉnh để ánh sáng rõ trên võng mạc. Tương tự, màu sắc và độ sáng tối của đối tượng cũng được cảm nhận như một hàng số, bất chấp sự thay đổi của độ sáng. Chính nhờ khả năng này mà hệ thống thị giác đối khi có thể giải quyết được vấn đề nhập nhằng như hình 1.2.



a) Chữ B hay số 13



b) Số 121314



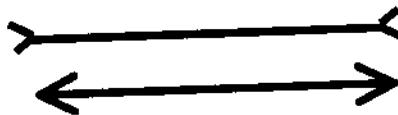
c) ABC

Hình 1.2.a biểu diễn chữ B hay con số 13? Tuy nhiên, trong hình 1.2.b, vẫn hình dạng đó song ta cảm nhận chính xác là con số 13; còn trong hình 1.2.c ta lại cảm nhận là chữ B. Như vậy sự cảm nhận còn phụ thuộc vào ngữ cảnh.

**Hình 1.2. Sự nhập nhằng và khả năng cảm nhận của hệ thống thị giác.**

Tuy nhiên, sự điều chỉnh này của hệ thống cảm nhận đôi khi dẫn đến ảo giác và bị đánh lừa. Như trong hình 1.3, cả 2 đoạn thẳng đều có độ dài bằng nhau, nhưng ta có cảm giác là đoạn thẳng ở trên dài hơn đoạn nằm phía dưới.

## Chương 1: Tâm lý nhận thức của con người trong giao tiếp và xử lý



Hình 1.3. Đoạn nào dài hơn?

Điều này xảy ra không chỉ đối với hình ảnh mà còn với cả văn bản. Các bạn dễ dàng tìm ra các ví dụ để minh họa.

Việc cảm nhận chữ viết trong quá trình đọc diễn ra trong nhiều giai đoạn. Trước tiên là quá trình nhận mẫu của từ trên trang. Tiếp sau là giai đoạn giải mã và đối sánh với sự biểu diễn bên trong của ngôn ngữ. Cuối cùng là giai đoạn xử lý ngôn ngữ với các phân tích về cú pháp và ngữ nghĩa được thực hiện trên từng mệnh đề hay từng câu. Trong ngữ cảnh tương tác người-máy, chúng ta chỉ quan tâm hai giai đoạn đầu.

Trong quá trình đọc, mắt người có thể chuyển động tiến hay lùi theo kiểu hồi qui. Văn bản càng phức tạp tính hồi quy càng tăng. Việc cảm nhận diễn ra trong chu kỳ cố định và chiếm khoảng 94% thời gian quan sát. Theo J. Dix [2], một người bình thường có thể đọc khoảng 200 từ/phút. Việc đọc không hoàn toàn tuân tự như máy quét vì con người có thể cảm nhận không chỉ ký tự mà cả từ. Với các từ quen thuộc hay khi chữ được viết rõ ràng, tốc độ đọc sẽ tăng lên.

Tốc độ đọc văn bản là số đo tính dễ đọc. Thực nghiệm chứng minh rằng, với phông chuẩn cỡ từ 9 - 12 cũng dễ đọc ti lệ với độ giãn cách dòng. Chiều dài dòng cũng ảnh hưởng đến tính dễ đọc. Đương nhiên là đọc trên màn hình máy tính chậm hơn là đọc trên sách.

Như vậy, có nhiều yếu tố ảnh hưởng đến tính dễ đọc và khi thiết kế giao tiếp cần chú ý là chữ đen trên nền trắng dễ đọc hơn là chữ trắng trên nền đen. Có nghĩa là độ tương phản ám tốt hơn độ tương phản dương.

### 1.2.2. Nghe

Thính giác được xem như giác quan thứ hai, song thông tin cảm nhận theo cách này trong tương tác người-máy là ít hơn. Cách thức cảm nhận thông tin theo kiểu này rất khác. Người ta có thể không nhìn mà chỉ qua nghe đã cảm nhận khá chính xác. Thí dụ như qua giọng nói hay âm thanh phát ra, ta đã có thể biết được đó là người quen hay ai, đối tượng phát ra đó là gì. Thậm chí có thể phán đoán cả hướng nơi phát ra âm thanh đó. Tóm lại, hệ thống thính giác có khả năng rất to lớn để truyền thông tin trong môi trường chúng ta. Vậy việc cảm nhận âm thanh diễn ra như thế nào?

Âm thanh là sự thay đổi hay rung động khi không khí bị nén. Âm thanh có nhiều đặc trưng khác nhau:

- Tần số
- Độ vang
- Âm sắc

Độ vang phụ thuộc vào độ khuếch đại, còn tần số là một hằng số. Âm sắc là phẩm chất đặc trưng cho âm thanh phát ra.

Tai người có thể nghe được âm thanh với tần số trong khoảng từ 20 Hz đến 15 kHz.

## Chương 1: Tâm lý nhận thức của con người trong giao tiếp và xử lý

Tai có thể phân biệt được sự thay đổi tần số với các thay đổi nhỏ hơn 1,5Hz ở tần số thấp, song lại kém chính xác ở tần số cao. Các tần số khác nhau sẽ kích thích các hoạt động trên các vùng khác nhau của bộ 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 các tiếng ồn để lấy các thông tin hữu ích.

Đúng là âm thanh có thể truyền đi một lượng thông tin đáng kể, song lại ít dùng trong thiết kế giao tiếp người - máy. Tuy nhiên, tai có thể phân biệt khá nhạy cảm với sự thay đổi của âm thanh và có thể nhận ra các âm thanh quen thuộc không cần chú ý đến nguồn gốc của nó. Điều này gợi ý cho các nhà thiết kế giao tiếp về sự mở rộng của âm thanh trong giao tiếp.

### *1.2.3. Nhấn phím*

Giác quan thứ ba chúng ta quan tâm là xúc giác (haptic perception). Xúc giác thực hiện bởi tác động sờ hay nhấn phím. Tuy việc cảm nhận thông qua giác quan này không quan trọng như hai giác quan trên trong đời sống hàng ngày, song nó lại cung cấp các thông tin có tính sống còn về môi trường của chúng ta, đặc biệt là trong tương tác người-máy.

Xúc giác cung cấp phương tiện chính trong hồi đáp nhất là khi sử dụng các hệ thống máy tính. Phải thừa nhận rằng, với người dùng máy tính, đó là nguồn thứ hai để cảm nhận thông tin, các nguồn khác là thứ yếu. Thí dụ như với hệ thống chữ nổi cho người mù thì phương tiện duy nhất là cảm giác qua tiếp xúc.

Hệ thống xúc giác khác hẳn với hệ thống thị giác và thính giác. Việc cảm nhận các kích thích thông qua làn da. Da có ba kiểu cảm nhận đặc biệt:

- Cảm nhận nhiệt
- Cảm nhận sức căng do áp suất
- Cảm nhận cơ khí.

Việc cảm nhận cơ khí lại phân ra hai dạng: dạng đáp ứng nhanh và loại đáp ứng chậm. Sức căng càng lớn thì phản ứng càng nhanh và có tính giật cục. Loại đáp ứng chậm lại có cảm nhận liên tục. Với người đánh máy hay người dùng máy tính sử dụng bàn phím, sự nhận thức vị trí tương đối của ngón tay và sự hồi đáp từ bàn phím là rất quan trọng.

### *1.2.4. Dịch chuyển*

Một hành động đơn giản như nhấn một phím lệnh để đáp ứng một yêu cầu bao gồm nhiều bước xử lý. Trước tiên, các kích thích được cảm nhận bởi hệ thống cảm nhận rồi truyền đến não. Yêu cầu được xử lý và một đáp ứng được khởi tạo. Tiếp theo, não sẽ gửi các tín hiệu đến các bộ phận tương ứng. Mỗi hành động cần một khoảng thời gian nhất định và chúng ta có thể coi nó gồm thời gian phản ứng và thời gian dịch chuyển.

Thời gian dịch chuyển phụ thuộc nhiều vào các đặc tính vật lý của chủ thể, thí dụ như tuổi tác và độ tinh tế, còn thời gian phản ứng lại thì rất khó đánh giá. Thời gian phản ứng phụ thuộc vào các giác quan mà kích thích tác động đến. Một người bình thường có thể phản ứng với tín hiệu âm thanh khoảng 150 ms, với một tín hiệu nhìn khoảng 200 ms và 700ms với một cảm giác đau đớn do va chạm. Có nhiều yếu tố ảnh hưởng đến thời gian

phản ứng. Thi dụ như nếu có kỹ năng và kinh nghiệm, thời gian phản ứng sẽ nhanh hơn, ngược lại khi một môi thời gian phản ứng sẽ chậm đi. Một câu hỏi đặt ra là liệu tốc độ phản ứng có dẫn đến độ chính xác giảm không, vì độ chính xác cũng là một số đo của dịch chuyên? Điều này còn phụ thuộc vào nhiệm vụ phải thi hành và người dùng. Trong một số trường hợp, nếu thời gian phản ứng nhanh thì độ chính xác sẽ giảm.

Tốc độ và độ chính xác của chuyển động là hai yếu tố rất quan trọng cần được xem xét khi thiết kế các hệ thống tương tác, nhất là khi cần xem thời gian chi phí cho việc di chuyển tới một đích cụ thể trên màn hình. Đích này có thể là một phím lệnh (command button), một biểu tượng hay một mục trên menu,... Thời gian chi phí này đã được Fitt nghiên cứu và mang tên ông “luật Fitt” như mô tả trong hình 1.4.

Giả sử rằng để chuyển đến đích, ta phải qua nhiều đích chuyển nhô (vi chuyển). Gọi:

D : Khoảng cách phải di chuyển

L: Độ rộng của đích

X<sub>i</sub>: Khoảng cách phải di chuyển sau i lần, X<sub>i</sub> =

$\varepsilon * X_{i-1}$ , với  $\varepsilon$  là hằng sai số:  $\varepsilon = X_i / X_{i-1}$ ,  $\varepsilon < 1$ .

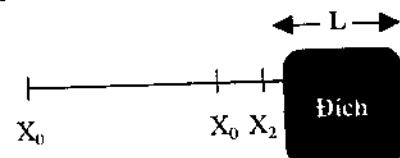
Vì  $X_0 = D$ , do vậy  $X_n = \varepsilon^n * D$ . Đích sẽ đạt được khi  $X_n \leq L/2$  hay  $\varepsilon^n * D \leq L/2$ .

Vì  $\log_2 \varepsilon^n = n \log_2 \varepsilon$  nên  $n = -\log_2(2D/L) / \log_2 \varepsilon$ .

Thời gian cần thiết để thực hiện 1 đích chuyển:

$t = ts + tc + tm$ , với

ts: bộ xử lý cảm nhận định vị tay, tc thời gian bộ xử lý nhận thức ra lệnh vi chuyển và tm: mô típ xử lý thực hiện vi chuyển.



$$T = I \log_2 2D/L \text{ với :}$$

$$I = -(ts + tc + tm) / \log_2 \varepsilon \approx 100 \text{ ms}$$

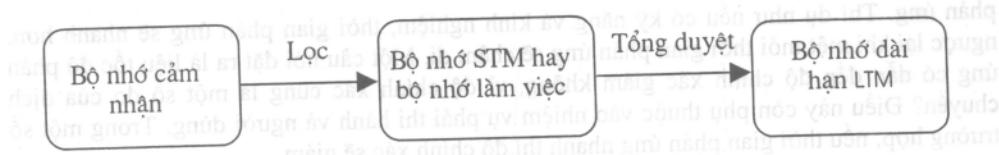
**Hình 1.4. Luật Fitt.**

Luật Fitt chỉ ra rằng thời gian chạm tới đích chỉ phụ thuộc vào tỉ số D/L, có nghĩa là phụ thuộc vào khoảng cách phải di chuyển và kích thước của đích. Luật này gợi ý rằng: để thao tác được thuận tiện và chính xác, cần thiết kế sao cho đích càng lớn và khoảng cách càng nhỏ thì càng tốt.

### 1.3. Bộ nhớ

Phần trên, chúng ta đã đề cập đến mô hình bộ xử lý thông tin con người của Card, Moral và Newell. Mô hình này bao gồm ba hệ thống con và mỗi hệ thống có bộ xử lý và bộ nhớ riêng. Trong phần này, chúng ta xem xét một cách chi tiết cấu trúc bộ nhớ và cách thức lưu trữ thông tin để hiểu bộ nhớ làm việc ra sao, tại sao chúng ta lại có thể nhớ được thông bất kỳ một cách nhanh chóng; tại sao người này nhớ tốt hơn người kia và tại sao chúng ta lại quên. Người ta cho rằng có ba kiểu bộ nhớ, đó là:

- Bộ nhớ cảm giác (sensory memory)
- Bộ nhớ ngắn hạn hay bộ nhớ làm việc (Short term memory)
- Bộ nhớ dài hạn (Long term memory).

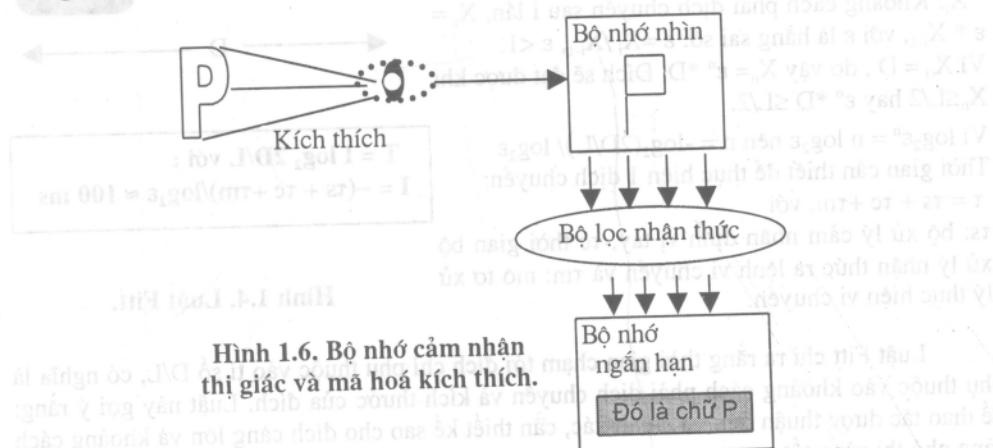


**Hình 1.5. Mô hình bộ nhớ.**

**1.3.1. Bộ nhớ cảm nhận**

Bộ nhớ cảm nhận chứa các kích thích nhận được từ các giác quan như nghe, nhìn, sờ. Người ta cho rằng, ứng với mỗi giác quan có một bộ nhớ cảm nhận riêng. Tại đây các kích thích được mã hoá. Hình 1.6 minh họa việc cảm nhận và mã hoá chữ P trong bộ nhớ cảm nhận.

Thông tin trong bộ nhớ cảm nhận được lưu theo cách viết đè: thông tin mới sẽ thay thế các thông tin cũ. Bộ nhớ cảm nhận có mối liên hệ mật thiết với bộ nhớ ngắn hạn của hệ thống nhận thức.



**Hình 1.6. Bộ nhớ cảm nhận, thị giác và mã hóa kích thích.**

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 (do dung lượng có hạn), thông tin không được chuyển sang nữa. Thông tin tồn lưu trong bộ nhớ cảm nhận với các chu trình khác nhau. Với bộ nhớ thị giác, thời gian tồn lưu khoảng 200ms, với bộ nhớ thính giác thì lâu hơn khoảng 1500ms.

### 1.3.2. Bộ nhớ ngắn hạn

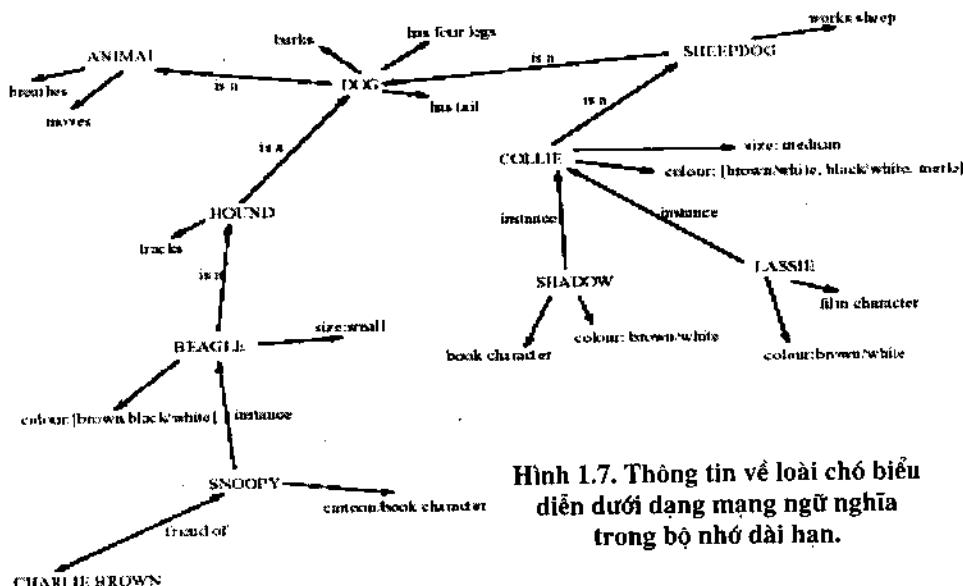
Bộ nhớ ngắn hạn tổ chức như kiểu thanh ghi của máy tính. Nó chứa các toán hạng vào hay các kết quả trung gian của các xử lý hiện thời. Các toán hạng đến từ bộ nhớ cảm nhận hay bộ nhớ dài hạn. Thông tin đến từ bộ nhớ cảm nhận không còn mang các đặc trưng vật lý nữa mà nó đã được giải nghĩa-các bản dịch. Thông tin đến từ bộ nhớ dài hạn được cắt thành từng đoạn (chunks), được kích hoạt bởi bộ xử lý nhận thức. Mỗi đoạn là một đơn vị nhận thức ký hiệu, một sự trùu tượng có thể kết hợp với các đơn vị khác. Sự kết hợp và bản chất của một đoạn phụ thuộc vào tri thức của một cá thể. Thí dụ, HUT là một chữ viết tắt và

với đa số người đều hiểu đó là tên Trường Đại học Bách khoa viết tắt bằng tiếng Anh. Tuy nhiên với nhiều người khác, họ sẽ hiểu là ba đoạn riêng biệt: H, U và T.

Bộ nhớ ngắn hạn có 3 đặc trưng chính. Đặc trưng thứ nhất là thời gian truy nhập nhanh, khoảng 70 ms. Tuy nhiên, thông tin hư hỏng cũng nhanh, khoảng 200 ms. Đó cũng là đặc trưng thứ hai. Đặc trưng thứ ba là dung lượng bộ nhớ hạn chế. Theo J.Dix, có 2 phương pháp để đo dung lượng bộ nhớ. Phương pháp thứ nhất dựa vào việc xác định chiều dài của một chuỗi mà ta có thể nhớ lại theo thứ tự. Phương pháp thứ hai dựa vào số mục mà ta có thể nhớ theo thứ tự bất kỳ. Theo công thức nổi tiếng của G.Miller, một người bình thường chỉ nhớ được khoảng  $7 \pm 2$  đoạn; có thể là dãy các con số, các sự kiện,... Khi mà bộ nhớ ngắn hạn bão hòa, sự kích hoạt của đoạn mới sẽ xoá đi thông tin trong bộ nhớ nếu nó không được nhắc lại. Điều này cũng gần giống chiến lược phân trang bộ nhớ của hệ điều hành. Điều này giải thích tại sao khi các con số hay sự kiện đến dồn dập ta chỉ nhớ được một số thôi, như theo công thức của Miller là khoảng từ 5 đến 9. Tuy nhiên, khi các đoạn được hình thành từ các mẫu tốt, khả năng nhớ sẽ tăng lên. Xét xâu chữ sau mà mỗi âm tiết hay đoạn gồm 2 hay 3 chữ cái: "hec atr unu pih etr eet". Rõ ràng là rất khó nhớ. Song nếu ta dịch chuyển chữ cái cuối cùng lên đầu và cũng tạo thành xâu mới theo quy tắc trên, ta có: "the cat run up the tree". Xâu này hiển nhiên là dễ đọc và dễ nhớ. Một thực tế khác cũng minh chứng rằng, kí tự cuối cùng trong một câu hay sự kiện cuối cùng trong một dãy là dễ nhớ nhất.

### 1.3.3. Bộ nhớ dài hạn

Bộ nhớ dài hạn có vai trò như bộ nhớ trung tâm và bộ nhớ thứ cấp của máy tính: nó lưu trữ thông tin, thông tin đó có thể được đọc hay bị thay đổi. Bộ nhớ dài hạn có một số đặc trưng chính sau: Thứ nhất nó có cấu trúc tuyền tính và có dung lượng lớn. Nói chung là dung lượng bộ nhớ này không hạn chế. Thứ hai là thời gian truy cập tương đối chậm, khoảng



**Hình 1.7. Thông tin về loài chó biểu diễn dưới dạng mang ngữ nghĩa trong bộ nhớ dài hạn.**

1/10 s. Thứ ba, thời gian hư hỏng chậm.

Về mặt cấu trúc, người ta cho rằng, bộ nhớ dài hạn gồm 2 kiểu: kiểu rời rạc (episode memory) và kiểu ngữ nghĩa. Bộ nhớ rời rạc chứa các biến cố và kinh nghiệm theo kiểu tuần tự. Chính nhờ bộ nhớ này, ta có thể xây dựng lại các biến cố theo các thời điểm khác nhau trong cuộc sống. Bộ nhớ ngữ nghĩa 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 có thể lấy từ bộ nhớ rời rạc và vì thế chúng ta có thể học các sự kiện mới hay các khái niệm từ kinh nghiệm. Thông tin trong bộ nhớ này được biểu diễn dưới dạng mạng ngữ nghĩa như chỉ ra trong hình 1.7.

Bộ nhớ ngữ nghĩa được cấu trúc theo cách thức cho phép ta truy nhập thông tin, biểu diễn mối quan hệ ngữ nghĩa giữa các mẫu tin và hỗ trợ suy diễn. Các mục được kết hợp với một lớp khác và có thể thừa kế các thuộc tính từ các lớp cha.

Hình 1.7 minh họa việc biểu diễn thông tin về loài chó bằng mạng ngữ nghĩa. Các thuộc tính về một nòi cụ thể được lưu cùng với nòi đó, do vậy thông tin chung về loài chó được lưu ở mức cao nhất. Điều này cho phép chúng ta khai quát hoá các trường hợp cụ thể. Thí dụ, chúng ta không thể nói loài chó chăn cừu Shadow có 4 chân và 1 đuôi, song có thể suy luận từ thông tin chung của nòi Sheepdog và Dog. Người ta cũng đề xuất một số cấu trúc khác của bộ nhớ trong việc biểu diễn tri thức và mỗi cấu trúc biểu diễn một sắc thái khác, thí dụ như frame (khung) hay Script. Dạng frame khá quen thuộc trong biểu diễn tri thức và thường được dùng để biểu diễn các đối tượng phức tạp. Frame có chứa các khe (slot) dùng để bổ sung giá trị của các thuộc tính. Các khe này có thể chứa giá trị ngầm định, cố định hay thay đổi như minh họa trong hình 1.8

DOG	Collie
Fixed	Fixed
Chân: 4	Nhánh của: chó
Default:	Loài: chó chăn cừu
Loài: ăn thịt	
Âm thanh: chóp tai	
Variable	Default:
Kích thước:	Kích thước: 65 cm
Màu sắc:	Variable
	Màu sắc:

**Hình 1.8. Biểu diễn tri thức dựa vào frame.**

Ngoài việc biểu diễn tri thức như trên, người ta còn sử dụng các luật kiểu điều kiện- hành động dạng “If cond then Action”, thí dụ như: “nếu chó trưởng thành thì nó sẽ chạy rông”.

Tiếp theo, chúng ta xem xét quá trình xử lý thông tin trong bộ nhớ dài hạn (LTM). Có ba loại hoạt động chính liên quan đến xử lý thông tin trong bộ nhớ LTM: lưu trữ/nhớ lại thông tin, quên và trích rút thông tin.

Thông tin từ bộ nhớ ngắn hạn chuyển đến bộ nhớ LTM theo cơ chế “tổng duyệt” (hình 1.5). Quá trình cũng có thể được tối ưu theo nhiều cách.

Về nguyên nhân thông tin bị quên có hai lý thuyết được nêu ra: thông tin bị hư hỏng hay có hiện tượng giao thoa. Với kinh nghiệm của mình, Ebbinhaus cho rằng thông tin bị hư hỏng theo thang lôgarit. Điều này có nghĩa là các thông tin mới hư hỏng nhanh hơn các thông tin cũ. Thuyết giao thoa cho rằng, khi chúng ta nhận thông tin mới sẽ gây nên sự mất mát thông tin cũ. Việc quên thông tin cũng còn do một nguyên nhân khác nữa là yếu tố xúc động. Người ta cho rằng các từ xúc động khó nhớ trong bộ nhớ ngắn hạn song lại dễ nhớ trong bộ nhớ LTM. Có khuynh hướng cho rằng các thông tin tích cực dễ nhớ hơn các thông tin tiêu cực.

Việc trích rút thông tin từ bộ nhớ LTM có thể chia thành hai kiểu: kiểu gọi lại (recall) và kiểu công nhận (recognition). Trong kiểu gọi lại, thông tin được tạo lại từ bộ nhớ; còn trong kiểu công nhận, sự hiện diện của thông tin cung cấp tri thức về nó mà chúng ta đã thấy từ trước. Sự công nhận là một hoạt động nhận thức kém phức tạp vì thông tin được cung cấp như một sự gợi ý. Tuy nhiên, sự gợi lại có thể trợ giúp bởi việc cung cấp các gợi ý trích rút và làm cho chủ thể truy nhập thông tin nhanh chóng. Một việc gợi ý như vậy có thể được tiến hành theo cách phân loại thông tin. Các thông tin trong một loại sẽ dễ nhớ hơn các loại khác.

### **1.4. Lập luận và giải quyết vấn đề**

Trên đây, chúng ta đã xem xét cách thông tin đến và đi cũng như cách lưu trữ thông tin của con người. Trong các phần dưới đây, chúng ta sẽ xem xét việc xử lý và quản lý thông tin. Có thể đó 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 dù. Suy nghĩ của con người là đơn giản và kiểu tự nhận biết: khi ta không luôn sẵn sàng xác định quá trình ta sử dụng, ta vẫn có thể xác định một cách đơn giản quá trình này.

Việc suy nghĩ đòi hỏi nhiều tri thức. Một số hoạt động tri thức là khá trực tiếp và các tri thức được yêu cầu có thể bị ràng buộc. Một số hoạt động khác lại yêu cầu một số lượng lớn tri thức từ các lĩnh vực khác nhau. Thí dụ, để thực hiện một phép tính, lượng thông tin yêu cầu là khá nhỏ từ một vài lĩnh vực liên quan; tuy nhiên, để để hiểu các tiêu đề trong một tờ báo lại đòi hỏi nhiều kiến thức về chính trị, cấu trúc xã hội, các sự kiện trên thế giới, ...

#### **1.4.1. Lập luận (reasoning)**

Lập luận là một quá trình, qua đó chúng ta sử dụng tri thức đã có để dựng nên kết luận hay suy diễn điều mới về lĩnh vực quan tâm. Con người hay sử dụng một vài kiểu lập luận như: suy luận, quy nạp hay phản chứng.

1. Lập luận kiểu suy luận: Lập luận kiểu này được tiến hành bằng cách đưa ra các kết luận cần thiết một cách lôgic từ các giả thiết. Các lập luận kiểu này có thể biểu diễn dưới dạng: “*Nếu điều kiện thì hành động...*”. Thí dụ, “*nếu là thứ sáu có ta sẽ đi làm*”. Cần chú ý là kết luận lô gich được dẫn xuất từ giả thiết và không nhất thiết phải đúng. Thí dụ: “*nếu trời mưa thì đất vẫn khô*”. Đây là suy luận hợp thức dù rằng nó mâu thuẫn với tri thức của chúng ta về cái đúng trong thế giới thực. Trong cách lập luận này, con người hay mang các tri thức

### **Chương I: Tâm lý nhận thức của con người trong giao tiếp và xử lý**

về thế giới vào quá trình suy luận. Đó là một cách suy luận tốt vì nó cho phép chúng ta tạo một lối đi tắt trong quá trình đối thoại và tương tác giữa con người. Nếu chúng ta quan tâm đến tính hợp lý hơn là chân lý thì mọi giả thiết phải được tường minh.

2. Suy luận quy nạp: Quy nạp là quá trình suy diễn từ cái đã biết sang cái chưa biết. Thí dụ như các con voi mà chúng ta đã nhìn thấy đều có vòi thì ta có thể kết luận là mọi con voi đều có vòi. Tất nhiên cách suy luận này là không tin cậy và có thể không chứng minh được là đúng. Chúng ta chỉ có thể chứng minh nó là sai. Tuy nhiên, thay vì không tin cậy, quy nạp là một quá trình hữu ích mà chúng ta sử dụng một cách không khác nhau trong việc học về môi trường của chúng ta. Thí dụ, chúng ta không bao giờ thấy hết mọi con voi đã từng sống hay sẽ chào đời, song chúng ta có một tri thức chắc chắn về các con voi mà chúng ta đã chuẩn bị cho chân lý với mọi mục đích thực tế, cái thường được suy luận một cách rộng rãi bằng quy nạp. Các nghiên cứu cho thấy con người thích suy luận tích cực hơn suy luận tiêu cực hay phủ định. Thí dụ, nếu ta thấy một con voi không có vòi, chúng ta vẫn không thích thay đổi khẳng định của mình là “mọi con voi đều có vòi”.

3. Phản chứng: Đây là cách suy luận thứ ba của con người. Đó là phương pháp chúng ta dùng để đưa ra các giải thích cho các sự kiện chúng ta quan sát. thí dụ, giả thiết rằng Sam thường lái xe rất nhanh khi say rượu, như vậy nếu bắt gặp Sam lái rất nhanh, chúng ta có thể khẳng định Sam bị say rượu. tuy nhiên, điều này rất không đáng tin cậy, bởi vì có thể có nhiều nguyên nhân khác. Như trong thí dụ trên, có thể vì vội mà Sam lái xe rất nhanh chứ không hẳn là say rượu.

Tuy nhiên, thay vì không đáng tin cậy, song con người lại hay thích suy luận như thế và cứ giữ chúng cho đến khi có một sự thật hiển nhiên để hỗ trợ một lý thuyết ngẫu nhiên hay một giải thích. Điều này có thể dẫn đến các vấn đề trong các hệ tương tác. Thí dụ, nếu một sự kiện luôn tiếp theo một hành động thì người dùng có thể cho là sự kiện luôn được tạo ra bởi hành động.

#### 1.4.2. Giải quyết vấn đề

Nếu lập luận là phương tiện để suy diễn thông tin mới từ cái đã biết thì giải quyết vấn đề là quá trình tìm lời giải cho một nhiệm vụ chưa biết bằng các tri thức chúng ta có. Giải quyết vấn đề của con người được đặc trưng bởi khả năng “thích nghi” thông tin đã có để xử lý thông tin mới. Có nhiều cách nhìn khác nhau về cách thức con người giải quyết vấn đề. Ra đời sớm nhất vào nửa đầu thế kỷ này là lý thuyết Gestalt nó bao gồm cả tính sử dụng lại các tri thức và cách suy nghĩ. Lý thuyết chủ yếu thứ hai được đề xuất bởi Newell và Simon vào những năm 1970. Đó là lý thuyết không gian bài toán. Lý thuyết này cho rằng bộ não là một bộ xử lý thông tin có giới hạn.

## 1. Lý thuyết Gestalt

Các nhà tâm lý học ở trường Gestalt cho rằng giải quyết vấn đề của con người bao gồm sản xuất và tái sản xuất. Tái sản xuất xây dựng từ các kinh nghiệm có trước như các hành vi được tuyên bố, còn sản xuất là dựng nên bản chất và cấu trúc hoá lại vấn đề. Các đích trùu tượng không đủ khả năng giải thích bản chất, không đủ khả năng suy luận để dẫn tới lời giải. Bản chất của lý thuyết này là dịch chuyển từ các hành vi đến lý thuyết xử lý thông tin.

Tuy rằng lý thuyết này khá hấp dẫn trong cách mô tả việc giải quyết vấn đề của con người song nó không cung cấp đủ các hiền nhiên hay cấu trúc để hỗ trợ lý thuyết.

## 2. Lý thuyết không gian bài toán

Newell và Simon giải định rằng giải quyết vấn đề tập trung vào không gian bài toán. Không gian bài toán bao gồm các phát biểu của bài toán và việc giải quyết vấn đề bao gồm việc khởi tạo các phát biểu này bằng cách dùng các phép dịch chuyển hợp lý. Bài toán có một trạng thái ban đầu và một trạng thái đích và người dùng sử dụng các phép toán để dịch chuyển từ cái đầu tiên đến cái đích cuối cùng. Tuy nhiên, không gian bài toán như vậy có thể là rất lớn và cần phải dùng heuristic để lựa chọn các phép toán thích hợp để đạt mục đích. Một heuristic quen thuộc là “means-ends analysis”- phân tích phương tiện-đích. Trong một heuristic như thế, trạng thái đầu tiên được sánh với trạng thái đích và một phép toán được chọn nhằm làm giảm bớt sự khác nhau giữa chúng. Thí dụ khi chúng ta bố trí lại các đồ vật trong phòng, trạng thái ban đầu hiển nhiên là trạng thái các đồ vật ở vị trí hiện tại, trạng thái cuối cùng là các đồ vật được bố trí lại, dịch chuyển vị trí. Sự khác nhau giữa hai trạng thái này là vị trí các đồ vật và chúng ta có thể sử dụng nhiều thao tác khác nhau như: bê, đẩy, . . . . Việc chọn thao tác nào là phụ thuộc vào kích thước, trọng lượng các đồ vật và chúng ta lại đặt ra các đích con mới.

Một đặc trưng quan trọng của mô hình Newell-Simon là các thao tác thực hiện bên trong các ràng buộc của hệ thống xử lý con người. Do vậy, việc thử kiểm không gian bài toán là hạn chế bởi dung lượng của bộ nhớ ngắn hạn và tốc độ mà thông tin được rút. Chú ý rằng kinh nghiệm sẽ cho phép ta giải quyết vấn đề một cách dễ dàng hơn khi chúng ta có thể cấu trúc không gian bài toán một cách tương ứng và chọn lựa các phép toán một cách hiệu quả.

Lý thuyết không gian bài toán và mô hình giải quyết bài toán tổng quát được ứng dụng rộng rãi trong giải quyết vấn đề trong các lĩnh vực được định nghĩa tốt.

## 1.5. Tâm lý và thiết kế hệ thống tương tác

Trong các phần trên của chương, chúng ta đã đề cập một cách tóm lược cách thức con người cảm nhận, xử lý, lưu trữ thông tin và cách suy luận, giải quyết vấn đề. Tuy nhiên áp dụng như thế nào những điều đã thấy vào việc thiết kế một hệ thống tương tác cũng có nhiều điều phải bàn. Đôi khi những kết luận trực tiếp có thể tạo dựng ngay. Thí dụ, chúng ta có thể suy diễn ngay rằng màu lơ không được dùng cho những chi tiết quan trọng bởi vì người dùng rất khó cảm nhận màu này.

Tuy nhiên, trong nhiều trường hợp, các ứng dụng không hiển nhiên hay đơn giản như vậy. Nó có thể nguy hiểm và dẫn chúng ta tạo nên các khái quát không hợp thức. Với mục đích áp dụng các nguyên tắc tâm lý học hoặc rút ra một tính chất trong thiết kế, chúng ta cần hiểu ngữ cảnh của nó theo cả 2 khái niệm: ở đâu nó phù hợp với các yếu tố tâm lý và các chi tiết của các kinh nghiệm thực tế, các biện pháp sử dụng và các khách thể có liên quan. Để trợ giúp cho các nhà thiết kế, các nguyên tắc và kết quả của các nghiên cứu trong tâm lý học đã đúc kết thành các chuẩn cho thiết kế, các mô hình và kỹ thuật đánh giá thiết kế.

Các kiến thức liên quan đến lĩnh vực này sẽ được trình bày chi tiết trong phần II và phần III của cuốn sách.

# 2

## MÁY TÍNH

### Nội dung

- 2.1. Tổng quan
- 2.2. Thiết bị vào
- 2.3. Thiết bị ra
- 2.4. Bộ nhớ
- 2.5. Xử lý thông tin
- 2.6. Kết luận

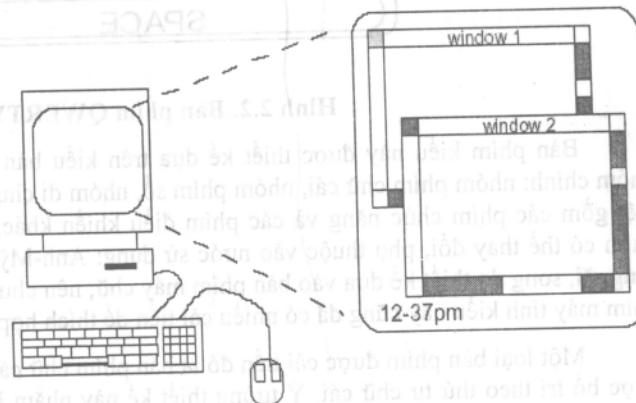
## 2.1. Tổng quan

Máy tính là thành phần thứ hai tham gia trong tương tác người-máy. Để hiểu cách con người tương tác với máy tính như thế nào, chúng ta cần phải hiểu rõ cả hai đối tượng. Trong chương Một, chúng ta đã tìm hiểu yếu tố con người và đã phân tích đặc điểm tâm lý nhận thức của người dùng trong ngữ cảnh tương tác. Trong chương này, chúng ta sẽ xem xét nhận thức của người dùng trong ngữ cảnh tương tác. Trong chương này, chúng ta sẽ xem xét nhận thức của người dùng trong ngữ cảnh tương tác. Trong chương này, chúng ta sẽ xem xét nhận thức của người dùng trong ngữ cảnh tương tác. Trong chương này, chúng ta sẽ xem xét nhận thức của người dùng trong ngữ cảnh tương tác.

Theo W. Thimbleby, một máy tính có thể được định nghĩa như sau: "máy tính là một thành phần tham gia tương tác, thực hiện một chương trình". Định nghĩa khái quát này có thể áp dụng cho nhiều thiết bị tương tác. Thí dụ, một công tắc đèn có thể được xem như một máy tính thực hiện một chương trình đơn giản: đèn sáng khi ta nhấn công tắc, đèn tắt khi ta nhả công tắc. Tất nhiên là thí dụ này quá đơn giản, song chúng ta cũng học được nhiều điều. Điều chính chúng ta quan tâm ở đây là một máy tính thực sự. Nó gồm những thành phần gì và vai trò của các thành phần đó trong giao tiếp?

Một máy tính tiêu biểu bao gồm các thành phần cơ bản như chỉ ra trong hình 2.1 dưới đây:

- Thiết bị vào ra (bao gồm vào ra văn bản và vào ra đồ họa): bàn phím, màn hình, chuột.
- Bộ nhớ: bộ nhớ ngắn hạn (RAM), bộ nhớ dài hạn: đĩa từ, đĩa CD.
- Bộ xử lý.



Hình 2.1. Các thành phần của một máy tính tiêu biểu.

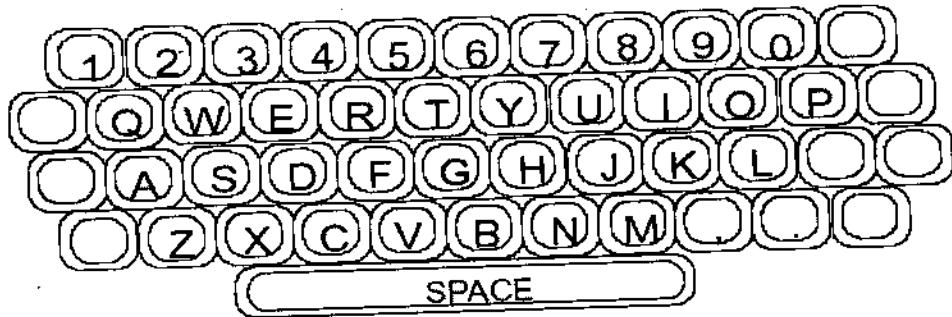
Trước khi xem xét chi tiết các thành phần, chú ý rằng máy tính có hai cách xử lý thông tin: theo lô (batch) hay trực tuyến (on-line). Thông tin vào theo lô ở đây là thông tin phải có khuôn dạng được định nghĩa tốt, sao cho người dùng có thể vào càng nhanh càng tốt. Với kiểu trực tuyến, người dùng ngồi trước máy tính và thông tin được đưa vào theo lời nhắc của máy. Theo cách này, máy tính được xem như một công cụ quản lý và nhận thông tin. Không một thông tin nào được bổ sung vào kho lưu trữ của máy tính, tuy nhiên thông tin được đưa vào theo dạng lệnh nhằm báo cho máy tính thực hiện một nhiệm vụ hay một tập nhiệm vụ chuyên dụng. Lĩnh vực tương tác người-máy tăng khi việc tương tác dùng máy tính tăng.

## 2.2. Thiết bị vào

Thiết bị vào là thiết bị chủ yếu cung cấp khả năng đưa thông tin cho máy tính. Các thiết bị này gồm hai loại chính: thiết bị vào văn bản và thiết bị định vị. Thiết bị vào văn bản gồm: các kiểu bàn phím QWERTY, DVORAK, ...; thiết bị định vị thường dùng trong chế độ đồ họa như: chuột, thiết bị định vị hai chiều.

### 2.2.1. Bàn phím

Bàn phím là thiết bị vào quan trọng và quen thuộc nhất hiện nay. Bàn phím được sử dụng chủ yếu để đưa vào dữ liệu dạng văn bản hay các câu lệnh. Các bàn phím được thiết kế theo các chuẩn. Loại thông dụng nhất là bàn phím QWERTY, minh họa qua hình 2.2.



Hình 2.2. Bàn phím QWERTY.

Bàn phím kiểu này được thiết kế dựa trên kiểu bàn phím máy chữ và bao gồm 4 nhóm chính: nhóm phím chữ cái, nhóm phím số, nhóm di chuyển con trỏ và nhóm phím đặc biệt gồm các phím chức năng và các phím điều khiển khác. Vị trí một vài phím trên bàn phím có thể thay đổi, phụ thuộc vào nước sử dụng: Anh-Mỹ hay Pháp. Tuy được sử dụng rộng rãi, song do thiết kế dựa vào bàn phím máy chữ, nên chưa thật tối ưu. Hiện nay các bàn phím máy tính kiểu này cũng đã có nhiều cải tiến để thích hợp hơn.

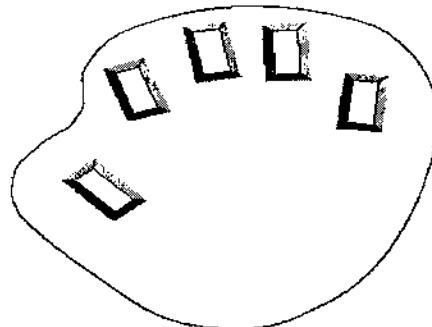
Một loại bàn phím được cài tiến đó là bàn phím chữ cái. Các phím trên bàn phím này được bố trí theo thứ tự chữ cái. Ý tưởng thiết kế này nhằm giảm thời gian cho những người sử dụng. Tuy nhiên, các nghiên cứu đã chỉ ra rằng loại bàn phím này không nhanh đối với những người dùng được đào tạo. Thiết bị này thường được sử dụng cho các thiết bị điện tử cá nhân bỏ túi.

Một loại bàn phím khác cũng được thiết kế và sử dụng: bàn phím DVORAK. Loại bàn phím này được thiết kế tương tự kiểu QWERTY, song các chữ cái được gán cho các phím khác nhau. Dựa vào việc phân tích cách gõ phím, bàn phím này được thiết kế nhằm giúp người dùng đạt tốc độ gõ phím cao. Nó được thiết kế trước tiên những người dùng thuận tay phải (56%). Việc thiết kế cũng làm giảm tối thiểu số lượng phím ở các vị trí ngón tay yếu. Thực nghiệm đã chứng minh rằng với bàn phím này tốc độ tăng từ 10 - 15% và giảm sự mệt mỏi của người dùng.

Một loại bàn phím khác cũng được thiết kế: bàn phím CHORD. Bàn phím này được thiết kế khá khác so với bàn phím thông thường (xem hình 2.3).

Loại bàn phím này chỉ có 4 hay 5 phím. Chữ cái được tạo ra khi nhấn đồng thời một số phím.

Bàn phím này có một số ưu điểm: kích thước nhỏ, học nhanh. Vì thế có thể áp dụng cho các máy tính bỏ túi hay các thiết bị điện tử. Tuy nhiên loại bàn phím này ít được ưa chuộng do tính thiếu tự nhiên của cách bố trí phím và số lượng phím.



Hình 2.3. Bàn phím CHORD.

#### 2.2.2. Một số thiết bị vào khác

Bên cạnh các loại bàn phím đã nghiên cứu ở trên, do đặc thù của các ứng dụng và cách thức tương tác, một số thiết bị vào khác cũng được nghiên cứu và thiết kế như: bộ nhận dạng chữ viết tay, nhận dạng tốc độ.

Chữ viết tay là một hoạt động khá quen thuộc và phổ biến. Nó cũng hấp dẫn như một phương pháp vào văn bản. Nếu chúng ta đã có khả năng viết như chúng ta viết trên giấy, thì với máy tính khi nhận dạng đầu vào kiểu này và biến đổi sang văn bản, chúng ta có thể thấy đó là một cách thức tương tác trực tiếp và đơn giản. Tuy nhiên có rất nhiều bất lợi với cách nhận dạng chữ viết. Thứ nhất, công nghệ hiện tại còn chưa đủ chính xác và còn gặp nhiều lỗi khi nhận dạng chữ. Thứ hai, các cá thể khác nhau trong nhận dạng rất lớn và gây nhiều khó khăn cho quá trình nhận dạng. Cái có ý nghĩa quan trọng nhất trong nhận dạng chữ không phải do chính dạng chữ mà do "stroke information" – cách mà chữ được vẽ lên. Điều này có nghĩa là thiết bị hỗ trợ quá trình nhận dạng phải hỗ trợ và tóm được thông tin về dáng nét chữ không phải là dạng ký tự cuối cùng. Điều này cũng có nghĩa là nhận dạng trực tuyến khó hơn nhiều so với đọc văn bản chữ viết trên giấy. Còn nhiều phức tạp khác khi mà các chữ cái trong từ là các dạng và thường được tạo nên rất khác nhau và phụ thuộc vào từ hiện tại; ngữ cảnh cũng có thể trợ giúp xác định việc đồng nhất chữ cái song thường không đủ khả năng cung cấp thông tin.

Có nhiều cách tiếp cận cho quá trình này: tiếp cận dựa vào bàn phím, tiếp cận dựa vào cách viết. Với cách tiếp cận dựa vào bàn phím, các hệ thống được thiết kế như các hệ thống tiêu biểu. Trong cách tiếp cận dựa vào cách viết, hệ thống nhận dạng được thị trường hóa một cách tích cực nhất là đối với các thiết bị bỏ túi. Các máy như thế được dùng chính để ghi chép, phác thảo ý tưởng như nhật ký, sổ địa chỉ.

Sử dụng nhận dạng chữ có nhiều ưu điểm so với dùng bàn phím. Hệ thống dựa vào cách viết thường nhỏ hơn, chính xác và dễ dùng hơn trong khi các phím nhỏ thì dễ mệt hoặc thậm chí không dùng được.

## Chương 2: Nhân tố máy tính trong tương tác người - máy

Ngoài chữ viết tay, tiếng nói cũng là một cách giao tiếp. Việc nhận dạng tiếng nói là một lĩnh vực khá hứa hẹn của nhận dạng văn bản. Tuy nhiên, cần phải có các thiết bị thích hợp và giá thành thường khá cao.

Ưu điểm của kiểu giao tiếp này là rất tự nhiên và tỉ lệ nhận dạng khá cao (90%). Tuy nhiên việc mở rộng hệ thống này với ngôn ngữ tự nhiên còn nhiều vấn đề phải giải quyết: lỗi chính tả, sự thiếu chính xác.

### 2.2.3. Các thiết bị trỏ và định vị

Với các máy tính hiện đại, cùng với các thiết bị vào văn bản như trên, các thiết bị trỏ và định vị cũng đóng một vai trò quan trọng trong giao tiếp người-máy. Các thiết bị này thường dùng để trỏ tới các đối tượng trên màn hình như một mục của menu, một folder, một bức ảnh, ..., hay thực hiện một tác động nào đó như định vị một vị trí trong một đoạn tài liệu, kích hoạt một phím lệnh. Chúng ta sẽ xem xét các thiết bị chính như thiết bị chuột, joystick,...

#### Chuột

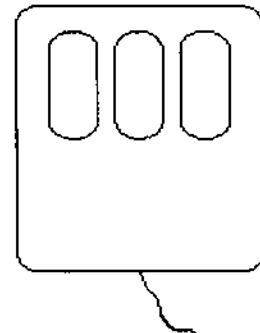
Chuột tuy xuất hiện sau bàn phím, song đã nhanh chóng trở thành thiết bị không thể thiếu của hầu hết các máy tính cá nhân, laptop, hay các trạm làm việc. Chuột thường được thiết kế với kích thước nhỏ và có từ 1 đến 3 phím (hình 2.4).

Khi hộp chuột di chuyển trên tấm trượt, viên bi xoay tròn bên trong hộp chạm vào các mấu (có 3 mấu) bô trí bên trong gây nên thay đổi về thế năng. Sự thay đổi về thế năng giúp đoán nhận sự thay đổi vị trí theo các hướng. Thông tin về sự chuyển động tương đối sẽ chuyển đến máy tính qua cáp nối nó với máy tính, và kết quả là con trỏ trên màn hình dịch chuyển theo.

Chuột một phím cũng được thiết kế như chuột nhiều phím, song loại này ngày nay không thông dụng. Các phím chuột được nhấn để thực hiện một hành vi lựa chọn hay khởi tạo một hành động. Tùy theo tình huống mà người ta nhấn đơn hay nhấn kép phím chuột. Cũng tương tự như trên với phím trái hay phím phải (với chuột ba phím, đó là phím trái cùng và phím phải cùng, phím giữa hầu như rất ít dùng).

Chuột là thiết bị vào gián tiếp và chỉ cung cấp thông tin về chuyển động tương đối của viên bi trong hộp. Nó có thể trượt từ góc trái trên màn hình đến tất cả mọi vị trí khác mà không tác động đến con trỏ.

Do cách di chuyển như vậy, thiết bị này không đòi hỏi nhiều không gian. Hơn nữa, do kích thước nhỏ và hình dáng đặc biệt nên người dùng không bị mỏi tay. Tuy nhiên do cách di chuyển như vậy nên cần có thời gian đào tạo, nhất là với những người dùng mới.



Hình 2.4. Thiết bị chuột.

### Một số thiết bị định vị hai chiều khác

Ngoài chuột, một số thiết bị khác có phương pháp tương tác tương tự cũng được thiết kế. Trước hết phải kể đến thiết bị Tracball. Nó là một thiết bị khá nhỏ, viên bi tròn chuyển động trong một hộp cố định. Việc phát hiện chuyển động cũng theo cùng cơ chế như chuột. Chuyển động tương đối của viên bi sẽ làm dịch chuyển con trỏ. Do cách bố trí và di chuyển như trên, tracball không cần không gian khi thao tác và thiết bị được thiết kế ở dạng nén. Nó cũng là thiết bị vào gián tiếp và yêu cầu sự phân tách các phím khi chọn. Về mặt chính xác, nó tốt hơn chuột, song lại tỏ ra có nhiều khiếm khuyết khi phải dịch chuyển dài. Tracball hiện nay được sản xuất theo nhiều kích thước song phổ biến như quả bóng trong trò chơi gôn. Nói chung, nó thích hợp với máy tính xách tay.

Joystick cũng là một thiết bị vào gián tiếp với kích thước khá nhỏ. Nó gồm một hộp mỏng với một cần. Nó là một thiết bị khá đơn giản. Sự chuyển động của cần tạo nên sự chuyển động tương đối ứng với con trỏ trên màn hình. Có hai loại joystick: loại tuyệt đối và loại đăng hướng. Với loại tuyệt đối, chuyển động là một đặc trưng quan trọng vì vị trí của cần tương ứng với vị trí con trỏ trên màn hình. Loại đăng hướng thì sức ép trên cần là quan trọng vì nó tạo nên một tốc độ (velocity) tương ứng của con trỏ và khi nó được thả cần lại về vị trí trung tâm. Joystick là một thiết bị không đắt và khá vững chắc, vì vậy nó thường được dùng trong các trò chơi điện tử.

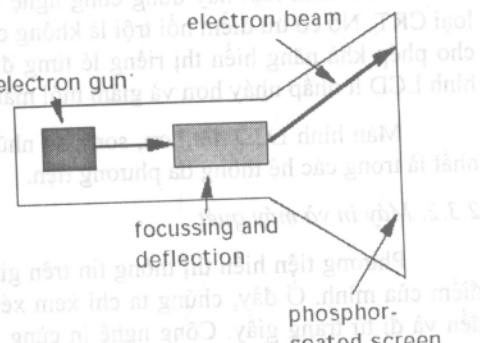
### 2.3. Thiết bị ra

Các thiết bị ra của máy tính khá phong phú. Nổi tiếng và quen thuộc nhất là màn hình. Tiếp theo là các thiết bị khác như máy in, loa,...

#### 2.3.1. Màn hình

Màn hình máy tính có nhiều dạng song phổ biến nhất là dạng ống điện tử CRT (Cathode Ray Tube). Hầu hết các hệ thống máy tính tương tác đều có màn hình sáng và là nơi hiện thông tin tương tác chủ yếu giữa người dùng và máy tính. Màn hình ống điện tử CRT được minh họa trên hình 2.5. Nguyên tắc của màn hình này khá đơn giản: một dòng điện tử bắn ra từ súng bắn điện tử. Dòng điện tử này được định hướng và điều khiển bởi từ trường, đập vào màn hình phủ photpho. Màn hình photpho khi bị các điện tử đập vào sẽ phát quang. Nguyên lý này được áp dụng cho hầu hết các thiết bị hiển thị.

Màn hình có 3 chế độ quét khác nhau: chế độ lưới (raster), chế độ ngẫu nhiên và chế độ trực tiếp.



Hình 2.5. Màn hình kiểu CRT.

#### Quét chế độ lưới

Quét lưới là dạng phổ biến nhất và là chuẩn cho các màn hình ti vi. Việc quét được thực hiện từ trái sang phải, từ trên xuống dưới với tần số quét là 30 Hz. Độ phân giải màn

## **Chương 2: Nhân tố máy tính trong tương tác người - máy**

---

hình trong chế độ này là  $640 \times 480$  điểm và có thể đạt tới  $1192 \times 980$  điểm. Màn hình đen trắng có thể hiện các ảnh đa cấp xám, màu hình màu hiện ảnh màu sử dụng ba màu cơ bản: đỏ (red), lục (green) và tím (blue) để hiện các màu khác nhau.

Màn hình CRT khá rẻ, thời gian đáp ứng nhanh và có khả năng hiện màu cao. Màn hình này cho phép các hàng có thể nghiêng tới  $45^\circ$ , nếu lớn hơn các nét sẽ bị gãy, không mịn.

### **Quét ngẫu nhiên**

Hiển thị ở chế độ ngẫu nhiên hay vector khác với chế độ lưới. Thay vì quét màn hình một cách liên tục từ trái sang phải và từ trên xuống dưới, quét ngẫu nhiên vẽ lên các đường cần hiển thị một cách trực tiếp. Việc quét nào được thực hiện ở bất kỳ vị trí nào trên màn hình và độ phân giải có thể đạt tới  $4096 \times 4095$ . Thiết bị hiển thị loại này đắt hơn so với thiết bị quét lưới và có thể áp dụng cho các ứng dụng “niche”.

### **Quét trực tiếp**

Là sự mở rộng của các thiết bị đo tín hiệu kiểu tương tự. Nó tương tự như chế độ ngẫu nhiên nhưng hình ảnh được duy trì bởi các “flood gun”, làm cho hình ảnh ổn định, không nhấp nháy. Độ phân giải có thể đạt tới  $4096 \times 3120$  điểm.

Một điều chú ý khi dùng màn hình hiển thị là người dùng thường cảm thấy mỏi mắt và mệt mỏi. Nguyên nhân chính là do sự nhấp nháy.

### **Màn hình tinh thể lỏng LCD**

Màn hình CRT có nhược điểm cồng kềnh, không thích hợp với các thiết bị yêu cầu gọn nhẹ như trong các máy tính xách tay hay một số thiết bị cá nhân khác. Việc sử dụng màn hình tinh thể lỏng là rất thích hợp với các thiết bị loại này. Loại thiết bị này thường dẹt và nhẹ với màu sắc thường xanh lá hay xám.

Màn hình loại này dùng công nghệ tinh thể lỏng, tiêu tốn ít năng lượng hơn so với loại CRT. Nó có ưu điểm nổi trội là không có phóng xạ. Do việc sử dụng ma trận địa chỉ, nó cho phép khả năng hiển thị riêng lẻ từng điểm ảnh. Do sử dụng cường độ sáng thấp, màn hình LCD ít nhấp nháy hơn và giảm mỏi mắt.

Màn hình LCD đắt hơn, song do những ưu điểm trên nên nó được sử dụng rộng rãi nhất là trong các hệ thống đa phương tiện.

#### **2.3.2. Máy in và máy quét**

Phương tiện hiển thị thông tin trên giấy có mặt từ rất lâu cùng với những ưu khuyết điểm của mình. Ở đây, chúng ta chỉ xem xét các công nghệ hiện tại để thu nhập thông tin đến và đi từ trang giấy. Công nghệ in cùng với các máy in sẽ được đề cập trước tiên, tiếp sau là máy quét và thiết bị nhận dạng ký tự quang học OCR.

#### **Máy in và việc in ấn**

Hệ thống máy tính dễ dàng tạo ra các văn bản trên giấy nhờ máy in. Có nhiều loại máy in khác nhau và chất lượng cũng rất khác nhau: từ máy in hàng đến máy in trắng, rồi

đến các máy in cao cấp hơn. Tuy nhiên cùng với sự giảm giá của các máy in và sự ra đời của các máy in lade và chất lượng cao, các máy in chất lượng thấp hầu như rất ít dùng.

Công nghệ in chính dựa vào khái niệm chấm đen (dot). Một ảnh hay một ký tự được tạo ra trên giấy bằng một chuỗi các chấm đen. Về mặt lý thuyết, mọi ký tự hay đồ họa đều có thể in được và chỉ bị giới hạn bởi độ phân giải của các chấm đen. Độ phân giải là một số đo được tính theo đơn vị dpi (dot per inch); số chấm đen trên một đơn vị chiều dài tính theo inches. Có ba loại máy in dựa vào ba công nghệ chấm đen: ma trận điểm, máy in phun và máy in lade.

### **Máy in ma trận điểm**

Các máy in loại này sử dụng một băng mực giống như máy chữ. Tuy nhiên, thay vì mỗi búa chỉ gõ một ký tự trên giấy, một hàng kim được dùng. Mỗi kim có thể đục một lỗ trên băng mực và tạo nên một dot trên giấy. Độ phân giải ngang có thể thay đổi khi điều chỉnh tốc độ các đầu trên giấy; còn độ phân giải dọc có thể được tăng lên nhờ gửi đầu kim hai lần đến các vị trí rất gần nhau. Vì vậy có thể điều chỉnh chất lượng in theo kiểu in thô (draft) hay chất lượng cao (fine).

### **Máy in phun**

Các máy in này hoạt động theo cách gửi các hạt mực từ hộp mực đến các đầu in lên giấy. Mực sẽ phun lên giấy do áp lực từ ink-jet, trong khi đó máy in bơm sử dụng bộ nung (heat) để tạo nên bột. Nhìn chung các máy in kiểu này không gây tiếng ồn. Độ phân giải của các máy in này tương tự như máy in ma trận điểm và có thể đạt tới 300 dpi.

### **Máy in lade**

Máy in này sử dụng công nghệ gần giống máy phô tô: các chấm đen của bộ tĩnh điện được phân bố trên mặt trống sẽ tạo nên chấm đen. Chấm này sẽ dán lên mặt giấy và được làm khô bởi bộ sấy. Độ phân giải của các máy in này khá cao, từ 600 đến 1200 dpi.

Các máy in này rất khác nhau về tốc độ và giá cả. Tốc độ các máy in tính theo số trang trên phút; với máy in kim là số ký tự. Tất nhiên là in một văn bản thuần túy sẽ nhanh hơn là văn bản có kèm đồ họa.

Trên đây, ta chỉ mới thuần túy xét đến công nghệ in. Còn những gì ảnh hưởng đến văn bản in? Các yếu tố này là phông in, cách thức mô tả trang in và sự tương ứng giữa trang in và trang màn hình.

### **Phông và ngôn ngữ mô tả trang in**

Phần lớn các máy in hoạt động theo cách thức: bất kỳ ký tự nào gửi đến máy in đều dưới dạng ASCII kiểu dáng máy chữ và trong dạng phông đơn. Theo lý thuyết, các ảnh cũng được in theo cách thức các chấm đen gửi tới máy in và không cần sự giải nghĩa nào cả. Trên thực tế, việc in ảnh không đơn giản như vậy.

Các văn bản in của ta thường rất phức tạp. Nó thường gồm nhiều đoạn văn bản với các phông chữ và kích cỡ khác nhau, với kiểu cách bình thường, nghiêng hay đậm, hay gạch chân. Trong văn bản thường có các đối tượng đồ họa, các ảnh số hoá, các tranh tạo ra bởi

## **Chương 2: Nhân tố máy tính trong tương tác người - máy**

---

các phần mềm hay từ thư viện tranh “Clip art”. Đôi khi máy tính phải làm mọi việc để chuyển đổi ảnh sang dạng bit map với kích cỡ phù hợp để gửi ra máy in. Thí dụ ở mức đơn giản nhất đó là các lệnh để thiết lập vị trí trang in, việc đổi cỡ phông in.

Văn bản in thường được đặc tả bởi một phông với một dạng và cỡ đặc biệt. Các dạng đó phân định đó là kiểu chữ có chân (serif) hay không chân (sans serif). Các định dạng đặc tả bởi tên phông. Cỡ phông được tính bằng điểm, một đơn vị có giá trị là 1/72 inches. Thí dụ cỡ chữ 12 có nghĩa là độ cao là 12/72, tương ứng với 6 dòng/inche. Một số phông có giãn cách (pitch) cố định như kiểu courier, một số khác lại thay đổi như Times-Roman...

Điều quan trọng khi sử dụng phông chữ trong giao tiếp người-máy đó là tính dễ đọc. Thực nghiệm chỉ ra rằng, văn bản chứa toàn chữ hoa khó đọc hơn văn bản toàn chữ thường. Điều thứ hai là cỡ chữ dễ đọc nằm trong khoảng 9 - 12. Tuy nhiên, một số chuẩn trong thiết kế giao tiếp lại cho rằng các chữ cái riêng lẻ và một số từ không có nghĩa sẽ dễ đọc hơn trong dạng chữ hoa.

### **Màn hình và trang in**

Một đòi hỏi chung nhất của các bộ xử lý văn bản và phần mềm chế bản đó là nguyên tắc WYSIWYG (What You See Is What You Get). Điều này có nghĩa là dáng vẻ của văn bản trên màn hình phải giữ nguyên khi hiện trên trang in. Nguyên tắc này với màn hình thì thỏa đáng, thí dụ như khi thao tác cẩn giữa thì văn bản xuất hiện ở giữa trang màn hình, song với phương tiện trang giấy lại có khác.

Trước tiên, đó là kích cỡ màn hình có khác kích cỡ trang giấy, ngay cả khi sử dụng khổ giấy A4 kiểu letter. Sự khác nhau này sẽ gây nên vấn đề khi thiết kế phần mềm. Liệu có thể làm cho kích cỡ màn hình giống như kích cỡ trang giấy không hay ngược lại? Nếu làm được, chỉ có thể đáp ứng được cái có thể hiển thị, còn lại các thứ khác sẽ bị lãng phí, thí dụ như độ phân giải. Một khó khăn nữa là vấn đề phông. Chúng ta hiện trên màn hình không giống như trên giấy: có thể rộng hơn hay hẹp hơn.

Sự khác nhau giữa màn hình và máy in nói lên rằng cần thiết kế riêng các mẫu đồ họa cho màn hình và cho máy in. Thí dụ, tiêu đề và các thay đổi để nhấn mạnh được thiết kế bằng cách dùng kiểu phông và cỡ trên giấy, việc sử dụng màu sắc, độ sáng tối hay các hộp thi như màn hình.

Nếu máy in nhận văn bản điện tử và chuyển nó lên giấy thì máy quét làm ngược lại quá trình này. Nó thực hiện bằng cách chuyển ảnh sang bitmap với sự giúp đỡ của thiết bị nhận dạng ký tự quang học. Thiết bị này lại chuyển đổi sang dạng ngược lại: văn bản. Ảnh hưởng của thiết bị này đến quá trình giao tiếp không nhiều, bạn đọc quan tâm xin xem tài liệu tham khảo.

### **2.4. Bộ nhớ**

Giống như bộ nhớ con người, bộ nhớ máy tính khi hoạt động ở các mức độ khác nhau được phân thành hai nhóm chủ yếu: bộ nhớ ngắn hạn và bộ nhớ dài hạn. Các thuật ngữ như dung lượng, tốc độ truy cập như đã trình bày trong chương một.

Việc chi tiết hoá bộ nhớ máy tính là không cần thiết vì nó không được quan tâm trực tiếp trong thiết kế giao tiếp người dùng. Tuy nhiên những giới hạn về dung lượng và các phương thức truy nhập là các ràng buộc quan trọng về các loại giao tiếp mà chúng ta có thể thiết kế.

### 2.4.1. Bộ nhớ ngắn hạn - STM

Bộ nhớ RAM được coi là bộ nhớ ngắn hạn của máy tính. Phần lớn các thông tin tích cực cần cho xử lý hiện thời được lưu trong RAM. RAM có nhiều loại khác nhau, khác nhau về thời gian truy nhập, cách tiêu thụ năng lượng và một số đặc trưng. Tốc độ truy nhập RAM nhanh, khoảng 100 ns và tốc độ truyền khoảng 10 MB/s. Dung lượng RAM thay đổi từ 32 MB tới 256 MB với các máy tính cá nhân.

Phần lớn các loại RAM là kiểu “volatile”, nghĩa là thông tin sẽ bị huỷ khi mất điện. Tuy nhiên, một số loại RAM khác lại thuộc kiểu “no volatile”. Loại này sử dụng một nguồn pin nhỏ để nuôi và làm tươi thông tin. Tất nhiên loại RAM này khá đắt. Thông tin từ bộ nhớ RAM được chuyển sang bộ nhớ dài hạn LTM.

### 2.4.2. Bộ nhớ dài hạn - LTM

Bộ nhớ dài hạn của máy tính là bộ nhớ ngoài như đĩa từ, đĩa CD... Nhìn chung, dung lượng của bộ nhớ này rất khác nhau: đĩa mềm 1,44 MB, đĩa cứng có thể đạt 40 GB hay hơn, còn đĩa CD khoảng 650 MB. Tốc độ truy nhập bộ nhớ này rất khác nhau và thường chậm so với bộ nhớ dài hạn.

## 2.5. Xử lý thông tin

Tốc độ xử lý có ảnh hưởng quan trọng đến giao tiếp người dùng. Những ảnh hưởng này cần được tính đến khi thiết kế các hệ thống tương tác. Có hai loại mà chúng ta cần xem xét đó là tốc độ quá chậm hoặc tốc độ quá nhanh.

Nếu tốc độ xử lý quá chậm, thông tin phản hồi từ hệ thống đến người dùng chậm, ảnh hưởng xấu đến quá trình giao tiếp. Thí dụ khi người dùng nhấn chuột trên một đối tượng: menu hay icon mà không thấy phản ứng (do hệ thống xử lý chậm) lại nhấn tiếp và có thể dẫn đến hiện tượng treo máy. Nếu tốc độ quá nhanh, người dùng không theo kịp và không biết hệ thống đã đáp ứng những gì.

Trong những trường hợp như vậy, người thiết kế cần cố định thời gian giữ chậm, phụ thuộc vào lớp người dùng và thiết bị.

Để thiết kế giao tiếp có hiệu quả, người thiết kế nên xem các giới hạn vật lý có thể như:

- Giới hạn của khả năng tính toán
- Giới hạn của kênh lưu trữ
- Giới hạn về khả năng đồ họa
- Dung lượng mạng khi thiết kế các phần mềm hoạt động trên môi trường mạng.

## 2.6. Tóm lược

Các thiết bị vào trình bày ở trên thực hiện hai chức năng chính là vào văn bản và

## ***Chương 2: Nhân tố máy tính trong tương tác người - máy***

---

định vị. Thiết bị vào chính là bàn phím kiểu QWERTY, tuy nhiên một số loại bàn phím khác cũng được thiết cho một số ứng dụng. Các thiết bị định vị, chủ yếu là thiết bị chuột cùng với một số thiết bị khác cũng được xem xét một cách chi tiết.

Các thiết bị ra, tập trung nhấn mạnh vào hai kiểu màn hình: loại màn hình CRT và loại LCD. Các đặc trưng về công nghệ sản xuất và tính năng được phân tích khá kỹ.

Tuy máy in không liên quan nhiều đến thiết kế giao tiếp, song cũng được xem xét. Bên cạnh đó, việc tương ứng giữa trang màn hình và trang in cũng được phân tích và đó cũng là điều cần chú ý trong thiết kế giao tiếp.

Điều quan trọng của chương này là người thiết kế cần có tri thức về tính năng của các loại thiết bị trên đó hệ thống được xây dựng. Nó không chỉ đơn thuần là các loại thiết bị vào ra mà mọi yếu tố có ảnh hưởng đến hành vi giao tiếp khi mà các yếu tố này ảnh hưởng đến bản chất và cách thức tương tác.

# 3

## CÁC MÔ HÌNH VÀ CÁC DẠNG TƯƠNG TÁC NGƯỜI - MÁY

### Nội dung

- 3.1. Tổng quan
- 3.2. Các mô hình tương tác
- 3.3. Các dạng tương tác
- 3.4. Tương tác WIMP
- 3.5. Điều khiển trực tiếp
- 3.6. Ngữ cảnh tương tác

### **3.1. Tổng quan**

Tương tác, trong ngữ cảnh ở đây, được hiểu là sự giao tiếp giữa người dùng con người và hệ thống, với máy tính được xem như một công cụ thực hiện, hay đơn giản là hỗ trợ nhiệm vụ. Có nhiều cách thức mà người dùng sử dụng để giao tiếp với hệ thống máy tính. Cách thức vào theo lô (batch input) là cách thức cổ điển nhất và thường được dùng trong các ngôn ngữ lập trình. Theo cách này, người dùng cung cấp mọi thông tin cho máy tính một lần và sau đó để cho máy tính thực hiện nhiệm vụ. Rõ ràng cách tiếp cận này liên quan rất ít đến tương tác người-máy và nó cũng không hỗ trợ bất cứ nhiệm vụ nào.

Một số cách tiếp cận khác là sử dụng các thiết bị vào - ra có tính tương tác cao và các mô thức, ví dụ như điều khiển trực tiếp. Ở đây, người dùng cung cấp các chỉ thị và nhận hồi đáp của hệ thống. Đó chính là các dạng tương tác mà chúng ta quan tâm.

Trong các phần dưới đây, chúng ta sẽ đề cập đến một số mô hình tương tác, các dạng tương tác được dùng và đánh giá mức độ hỗ trợ người dùng của nó như thế nào.

### **3.2. Các mô hình tương tác**

Tương tác người-máy bao gồm hai thành phần tham gia: người dùng và máy tính. Cả hai thành phần này đều rất phức tạp như đã đề cập trong chương một và chương hai. Hai thành phần này rất khác nhau trong cách thức giao tiếp. Để hiểu được một cách chính xác cái gì sẽ diễn ra trên tương tác và hiểu được gốc rễ của những khó khăn, người ta hay dùng mô hình để thực hiện. Chúng ta sẽ xem xét hai mô hình: mô hình “chu trình thực hiện - đánh giá” của D. Norman và mô hình “khung làm việc” (framework), một sự mở rộng của mô hình trên, phát triển bởi Abowd và Beale.

#### *3.2.1. Mô hình chu trình thực hiện - đánh giá*

Mô hình của Norman rất gần với sự hiểu biết cụ thể của chúng ta về tương tác giữa người dùng con người và máy tính. Người dùng hình thành một kế hoạch hành động và sau đó nó được thực hiện bởi giao diện máy tính. Khi kế hoạch hay một phần kế hoạch được thực hiện, người dùng quan sát kết quả trả về qua giao tiếp và quyết định các hành động tiếp theo.

Chu trình tương tác có thể chia hai giai đoạn chính: thực hiện và đánh giá. Các giai đoạn này lại có thể chia thành các bước nhỏ hơn:

- |                         |   |
|-------------------------|---|
| 1- Thiết lập mục đích   | 5- Cảm nhận trạng thái hệ thống               |
| 2- Hình thành chủ ý     | 6- Phân tích trạng thái hệ thống              |
| 3- Đặc tả dãy hành động | 7- Đánh giá trạng thái hệ thống với mục đích. |
| 4- Thực hiện hành động  |   |

Mỗi bước là một hành động của người dùng. Trước tiên, người dùng xác lập mục đích là những cái cần thực hiện. Tuy nhiên, trong bước này, mục đích mới còn ở mức khái quát, cần tiến hành làm rõ hơn như đặc tả các hành động trước khi nó được thực hiện. Người dùng nhận các trạng thái mới của hệ thống sau khi thực hiện một dãy hành động và phân tích theo chủ ý của mình hy vọng nó sẽ đạt mục đích. Nếu các trạng thái hệ thống phản ánh đúng mục đích, hệ thống đã thực hiện điều mà người dùng mong muốn và sự tương tác

đã thành công. Ngược lại, người dùng phải hình thành đích mới và khởi tạo lại chu trình thực hiện hành động.

Chú ý rằng, người dùng và hệ thống sử dụng ngôn ngữ riêng của mình. Hệ thống sử dụng ngôn ngữ khung (core language), còn người dùng sử dụng ngôn ngữ nhiệm vụ (task language).

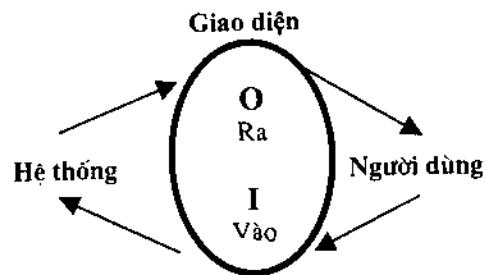
Mô hình của Norman là một phương tiện hữu ích để hiểu tương tác: nó đơn giản và cụ thể. Tuy nhiên, mô hình này mới chỉ xem xét hệ thống như là một giao diện theo quan điểm của người dùng mà không chú ý đến giao tiếp của hệ thống qua tương tác.

### 3.2.2. Mô hình khung tương tác

Mô hình này được Abowd và Beale phát triển dựa trên ý tưởng của Norman, nhằm cung cấp một mô tả hiện thực hơn khi mô tả hệ thống một cách tường minh. Theo mô hình này, hệ thống gồm 4 phần như minh họa trên hình 3.1.

- Hệ thống
- Người dùng
- Đầu vào
- Đầu ra

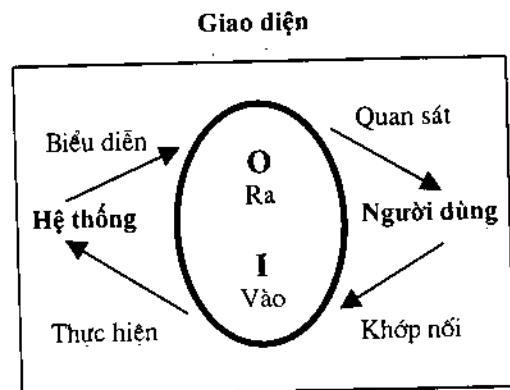
Mỗi thành phần sử dụng ngôn ngữ riêng của mình: hệ thống dùng ngôn ngữ core, người dùng dùng ngôn ngữ nhiệm vụ. Thành phần vào ra được biểu diễn tách riêng và phối hợp cùng nhau trên giao tiếp.



Hình 3.1. Mô hình framework.

Vì giao diện ở giữa hai thành phần: người dùng và máy tính, có 4 bước trong chu trình tương tác và mỗi bước tương ứng với một dịch chuyển từ thành phần này tới thành phần kia như trong hình 3.2.

Việc hình thành nhiệm vụ nhằm thực hiện mục đích cần được khớp nối trong ngôn ngữ đầu vào. Các nhiệm vụ này phải được dịch chuyển thành các kích thích đầu vào. Việc khớp nối được xem xét trong thuật ngữ bao trùm các nhiệm vụ đầu vào và phải khá dễ cho công đoạn dịch chuyển. Nhiệm vụ được nhấn mạnh bởi một số thuật ngữ có các thuộc tính tâm lý để chỉ rõ các đặc trưng quan trọng trong lĩnh vực đối với người dùng. Nếu các thuộc tính tâm lý này ánh xạ một cách rõ ràng sang ngôn ngữ đầu vào thì việc khớp nối các nhiệm vụ sẽ đơn giản hơn nhiều.



Hình 3.2. Tương tác người dùng-máy tính qua mô hình framework.

### **Chương 3: Các mô hình và các dạng tương tác người máy**

---

Ở giai đoạn tiếp theo, các đáp ứng của đầu vào sẽ được dịch chuyển để kích thích hệ thống. Điều quan tâm trong sự dịch chuyển này là liệu ngôn ngữ dịch chuyển đầu vào có thể đạt tới một số trạng thái của hệ thống như khi sử dụng các kích thích trực tiếp.

Khi một dịch chuyển trạng thái xảy ra bên trong hệ thống, giai đoạn thực hiện tương tác được hoàn thành và giai đoạn đánh giá bắt đầu. Trạng thái mới của hệ thống cần được thông báo cho người dùng và việc này bắt đầu bằng việc dịch chuyển đáp ứng của hệ thống thành các kích thích trên thành phần đầu ra. Việc biểu diễn dịch chuyển này phải bảo vệ các thuộc tính hệ thống có liên quan của lĩnh vực trong giới hạn biểu diễn của các thiết bị ra. Khả năng để nắm bắt các khái niệm trong lĩnh vực của hệ thống đầu ra là một vấn đề của tính biểu diễn được đối với sự dịch chuyển này.

Khung tương tác được thể hiện như một công cụ nhằm phán xét tính hiệu dụng của một hệ thống tương tác hoàn chỉnh. Thực tế, mọi việc phân tích được gọi ý bởi khung tương tác là độc lập trên nhiệm vụ hiện thời, qua đó người dùng thực hiện tương tác. Điều này không có gì ngạc nhiên bởi vì chỉ khi thực hiện một nhiệm vụ cụ thể trong một lĩnh vực nào đó, chúng ta mới có khả năng xác định các công cụ chúng ta sử dụng có đáp ứng không? Thí dụ, các hệ soạn thảo khác nhau là tốt cho các nhiệm vụ khác nhau. Hệ soạn thảo tốt nhất, nếu chúng ta bắt buộc phải chọn một, đó là cái phù hợp nhất cho nhiệm vụ thường xuyên phải thực hiện.

#### **3.3. Các dạng tương tác**

Việc tương tác được xem như sự đối thoại giữa người dùng và máy tính. Việc lựa chọn kiểu giao tiếp sẽ ảnh hưởng sâu sắc đến bản chất của quá trình đối thoại. Có nhiều kiểu tương tác được sử dụng để thiết kế giao tiếp như:

- Giao tiếp dòng lệnh
- Giao tiếp bằng chọn (menu)
- Giao tiếp bằng ngôn ngữ tự nhiên
- Giao tiếp dạng hỏi đáp và truy vấn
- Giao tiếp dạng form điền
- Giao tiếp dạng WIMP

Trong các phần dưới đây, chúng ta sẽ trình bày chi tiết các dạng giao tiếp cùng với các ưu, khuyết và cách dùng.

##### **3.3.1. Giao tiếp dòng lệnh**

Giao tiếp dòng lệnh là một loại giao tiếp có tính lịch sử và rất phổ biến (hình 3.3). Loại giao tiếp này cung cấp phương tiện biểu diễn dòng lệnh cho máy tính một cách trực tiếp. Người dùng đưa vào dòng lệnh để thực hiện yêu cầu của mình bằng cách nhấn một phím chức năng, phím tắt hay sử dụng các phím riêng lẻ. Các lệnh thường là các động từ viết tắt hay chọn từ.

Trong một vài hệ thống như MS DOS, UNIX, giao tiếp dòng lệnh là phương tiện giao tiếp cơ bản nhất giữa người dùng và máy tính. Ưu điểm chính của giao tiếp này là đơn giản, nhanh, tiện lợi và tốn thấp do cách nhập trực tiếp. Với người dùng có kinh nghiệm hay

### Chương 3: Các mô hình và các dạng tương tác người máy

các chuyên gia, đây là kiểu giao tiếp nhanh nhất. Kiểu giao tiếp này khá mềm dẻo vì các lệnh thường có các tuỳ chọn, và người dùng có thể sử dụng trong các tình huống khác nhau.

```
C:\WINDOWS\Desktop>dir
 Volume in drive C has 1000000 bytes free
 Volume Serial Number is 0000-0000
 Directory of C:\WINDOWS\Desktop

12/01/98  12:17a  AutoCAD R14.lnk
12/01/98  12:15a Microsoft Word (2).lnk
12/01/98  2:24a Winzip32.lnk
12/01/98  12:14a Shortcut to Menu.pif
12/01/98  12:02a Vietkey 2000.LNK
12/01/98  1:01a Shortcut to LINES.pif
12/01/98  12:01a Shortcut to Bkav401.lnk
12/01/98  12:01a Shortcut to 3½ Floppy
12/01/98  12:06a Shortcut (2) to Norton
12/01/98  12:28a Microsoft Visual Foxpro
12/01/98  12:03a Acrobat Reader 5.0.1
12/01/98  12:29a Shortcut to CD-ROM Drive
12/01/98  12:07a MSDN Library - April
               18 file(s)      7,526 bytes
               2 dir(s)    226,127,872 bytes free
```

Hình 3.3. Giao tiếp kiểu dòng lệnh.

Tuy nhiên, trở ngại lớn nhất của kiểu giao tiếp này là người dùng bắt buộc phải nhớ lệnh và cú pháp lệnh chứ không thể nhập tuỳ tiện. Điều này là khó với người mới dùng và ngay cả với người dùng có kinh nghiệm khi câu lệnh quá dài. Hệ quả trực tiếp là phải tốn công đào tạo. Các khó khăn này có thể được giảm nhẹ khi dùng các từ có nghĩa, gần với ngôn ngữ người dùng hơn là các thuật ngữ kỹ thuật.

Tuy vậy, cũng rất khó cho các người dùng vì các hệ thống khác nhau có thể sử dụng theo nghĩa khác nhau, gây nên sự nhầm lẫn trong giao tiếp cũng như học sử dụng.

#### 3.3.2. Giao tiếp kiểu bảng chọn

Cách thức giao tiếp này, cung cấp một tập các lựa chọn có thể cho người dùng và tập này được hiện trên màn hình (hình 3.4).

<b>PAYMENT DETAILS</b>	<b>PAYMENT DETAILS</b>
Please select payment method:	
1. Cash	
2. Cheque	
3. Credit card	
4. Invoice	
Enter a number for your choice	

Hình 3.4. a) Giao tiếp kiểu bảng chọn đơn giản.      b) Mục Cheque được chọn.

### ***Chương 3: Các mô hình và các dạng tương tác người máy***

Người dùng lựa chọn một mục (tương ứng một công việc) bằng cách sử dụng các phím con trỏ, phím tắt hay nhấn vào một ký tự (chữ hay số). Nếu chuột có thiết lập, người dùng có thể sử dụng trong thao tác lựa chọn các mục.

Khi các lựa chọn được quan sát trực tiếp trên màn hình, người dùng được gọi ý và không đòi hỏi phải nhớ. Kiểu bảng chọn này không chỉ sử dụng trong môi trường văn bản mà có thể sử dụng cả trong môi trường đồ họa.

Bảng chọn có thể tổ chức theo kiểu phân cấp, số mức không hạn chế. Rõ ràng cách giao tiếp này rất thích hợp cho người dùng không thường xuyên và tốn ít công đào tạo. Tuy nhiên giao tiếp kiểu bảng chọn cũng thể hiện một số nhược điểm như chiếm không gian màn hình, nhất là khi có quá nhiều lựa chọn với nhiều cấp độ khác nhau, người dùng khó có thể nhớ điểm xuất phát. Nhược điểm này có thể khắc phục bằng cách nhóm các lựa chọn theo chức năng, thêm mục trống hay tạo đường kẻ ngang như thường thấy trong các ứng dụng soạn thảo văn bản, quản trị dữ liệu, ...

Một điểm yếu nữa là tính thiếu mềm dẻo. Người dùng không thể cung cấp gì thêm ngoài các lựa chọn đã liệt kê và thường chậm với người dùng thành thạo.

#### ***3.3.3. Giao tiếp bằng ngôn ngữ tự nhiên***

Giao tiếp bằng ngôn ngữ tự nhiên là một dạng hấp dẫn nhất giữa người dùng và máy tính. Việc hiểu ngôn ngữ tự nhiên bao gồm cả tiếng nói và chữ viết, là một chủ đề được quan tâm và nghiên cứu của nhiều lĩnh vực. Tuy nhiên, sự nhập nhằng của ngôn ngữ tự nhiên gây nên các khó hiểu cho máy. Sự nhập nhằng của ngôn ngữ tự nhiên thể hiện ở nhiều mức độ: cú pháp, câu trúc, ..., làm cho một câu có thể không rõ ràng. Thí dụ câu: "The man hit the boy with the stick", có thể hiểu theo hai nghĩa: người đàn ông đánh đứa trẻ bằng gậy hay người đàn ông đánh đứa trẻ có gậy.

Ngay cả khi cấu trúc câu rõ ràng thì vẫn có thể tồn tại sự không rõ ràng trong nghĩa của từ. Con người thường dựa vào ngữ cảnh để phân loại các nhập nhằng này. Tuy nhiên, điều này với máy thì quả là khó, và có nghĩa là giao tiếp dựa vào ngôn ngữ tự nhiên là không có thể. Tuy nhiên, chúng ta hy vọng có thể xây dựng được một hệ thống dựa vào tập con của ngôn ngữ. Việc sử dụng ngôn ngữ tự nhiên trong một số lĩnh vực là có thể.

#### ***3.3.4. Giao tiếp dạng hỏi đáp và truy vấn***

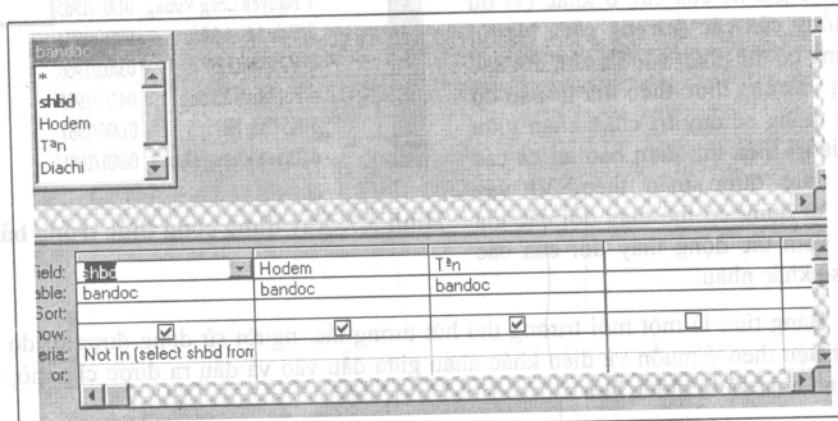
Hỏi đáp là một cơ chế đơn giản nhằm cung cấp dữ liệu cho một ứng dụng của một lĩnh vực riêng nào đó. Người dùng được yêu cầu bởi một loạt câu hỏi. Các câu hỏi được miêu tả trong nhiều dạng khác nhau: dạng Yes/No, dạng đa lựa chọn, dạng nhấn số. Việc tương tác này được tiến hành qua nhiều bước.

Kiểu giao tiếp này khá tự nhiên, dễ thiết kế và rất thích hợp cho người dùng mới và thiếu kinh nghiệm.

Các ngôn ngữ truy vấn, mặt khác được sử dụng để xây dựng các truy vấn lấy thông tin từ một cơ sở dữ liệu. Chúng sử dụng các cách diễn đạt kiểu ngôn ngữ tự nhiên, nhưng thực tế lại yêu cầu cú pháp rõ ràng, cũng như hiểu biết về cấu trúc cơ sở dữ liệu. Các truy vấn thường yêu cầu người sử dụng chỉ rõ một thuộc tính hay nhiều thuộc tính để tìm trong

cơ sở dữ liệu cũng như thuộc tính quan tâm để thực hiện. Điều này dễ hiểu khi có một thuộc tính đơn, nhưng trở nên phức tạp khi có nhiều thuộc tính được đề cập, ví dụ nếu thuộc tính A hoặc thuộc tính B làm cho người sử dụng chú ý hay thuộc tính A và phủ định thuộc tính B hay những giá trị các thuộc tính được so sánh. Hầu hết các ngôn ngữ truy vấn không cung cấp trực tiếp sự xác nhận những cái gì đã được yêu cầu, vì vậy chỉ xác nhận tính hợp lệ mới làm cho người sử dụng có kết quả tìm kiếm. Vì vậy, để sử dụng hiệu quả các ngôn ngữ truy vấn yêu cầu phải có một chút kinh nghiệm.

Nhiều ngôn ngữ truy vấn đã có cải tiến đáng kể bằng cách cung cấp giao diện đồ họa để người dùng điền thông tin vào. Việc sinh ra câu lệnh với cú pháp đầy đủ được thực hiện bởi phần mềm và trong suốt với người dùng (hình 3.5)



Hình 3.5. Giao tiếp dùng ngôn ngữ truy vấn sử dụng môi trường đồ họa.

### 3.3.5. Giao tiếp điền theo mẫu

Hệ thống hiển thị một tập các trường văn bản trên màn hình, người dùng có thể chọn một trường nào đó và nhập hoặc hiệu chỉnh nội dung của trường đó. Các giao diện điền theo mẫu được sử dụng chủ yếu cho nhập dữ liệu, nhưng cũng hữu ích trong các ứng dụng phục hồi dữ liệu. Người sử dụng được giới thiệu với một hiển thị giống như tờ mẫu có các ô trống để điền.

Thường các mẫu hiển thị dựa trên mẫu thực tế với những gì mà người sử dụng quen thuộc nhằm tạo nên giao diện dễ dàng hơn cho người sử dụng.

Người sử dụng làm việc xuyên suốt mẫu, điền các giá trị thích hợp. Dữ liệu được nhập vào ứng dụng ở vị trí chính xác. Hầu hết các giao diện mẫu điền cho phép dễ dàng di

### Hoá đơn đặt chỗ du lịch nhanh

Nhập vào chi tiết hành trình:

Nơi xuất phát: **Hà Nội**

Nơi đến: **Vũng Tàu**

Đường đi: **Đường số 1**

Số chỗ:

Hình 3.6. Giao tiếp điền mẫu đơn giản.

### **Chương 3: Các mô hình và các dạng tương tác người máy**

chuyên vòng quanh mẫu và cho phép một số trường để trống. Chúng cũng yêu cầu dễ dàng hiệu chỉnh vì người sử dụng có thể thay đổi ý định của họ hoặc gây ra lỗi giá trị nhập trong mỗi trường. Dạng hội thoại là chủ yếu hữu ích cho các ứng dụng vào dữ liệu. Ở một chừng mực nào đó, nó dễ cho việc học và sử dụng đối với những người sử dụng không thành thạo. Tuy nhiên, do có một thiết kế đầu vào linh hoạt, mẫu điện cũng thích hợp với những người sử dụng chuyên nghiệp.

Bảng tính là sự biến đổi tính vi của mẫu điện. Bảng tính bao gồm một lưới các ô, mỗi ô có thể chứa một giá trị hay công thức. Công thức có thể bao gồm các giá trị của các ô khác (ví dụ tính tổng của các ô trong cột). Người sử dụng có thể nhập vào và sửa đổi các giá trị và công thức theo thứ tự nào đó và hệ thống sẽ duy trì chắc chắn giữa các giá trị hiển thị, đảm bảo tất cả các công thức được tuân theo. Vì vậy người sử dụng có thể thay đổi các giá trị để xem tác động thay đổi của các tham số khác nhau.

Bảng tính là một môi trường thu hút tương tác: người sử dụng được tự do thao tác các dữ liệu theo ý muốn và điều khác nhau giữa đầu vào và đầu ra được che mờ, tạo giao diện linh hoạt và tự nhiên hơn.

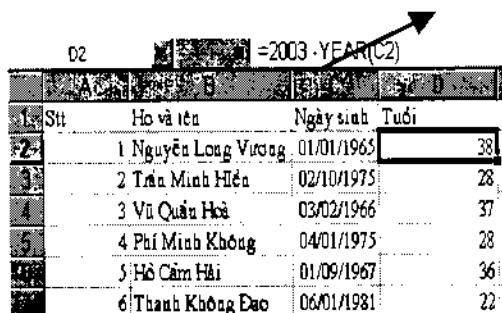
#### **3.4. Tương tác WIMP**

Hiện nay hầu hết các môi trường chung cho tương tác máy tính là các ví dụ dạng giao diện WIMP, thường được gọi là các hệ thống cửa sổ đơn giản. WIMP thay thế cho các cửa sổ, các biểu tượng, các menu và các con trỏ (thỉnh thoảng là các cửa sổ, các biểu tượng, chuột và các menu đồ xuống), và là dạng tương tác mặc định cho phần lớn hệ thống tương tác máy tính đang sử dụng hiện nay, đặc biệt là ở máy PC và máy trạm. Các ví dụ về các giao diện WIMP bao gồm Microsoft Windows cho máy IBM PC để bàn, MacOS cho máy Apple Macintosh để bàn và các hệ thống X Windows-based cho UNIX.

Chúng ta hầu như đã chú ý đến bốn đặc trưng then chốt của giao diện WIMP bao gồm: windows, icons, menus và pointers, và bây giờ chúng ta sẽ mô tả lần lượt. Đồng thời các thành phần này của các giao diện WIMP được gọi là các widget và chúng bao gồm bộ dụng cụ cho tương tác giữa người sử dụng và hệ thống.

##### **3.4.1. Windows**

Các cửa sổ là các vùng của màn hình, chúng có thể được xem như một màn hình độc lập. Một cửa sổ thường chứa văn bản hoặc đồ họa và có thể được di chuyển hay chỉnh lại kích thước theo ý muốn trong giới hạn cho phép. 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. Người sử dụng có thể thao tác trên các cửa sổ khác nhau, ví dụ họ có thể chuyển từ một dòng công việc này



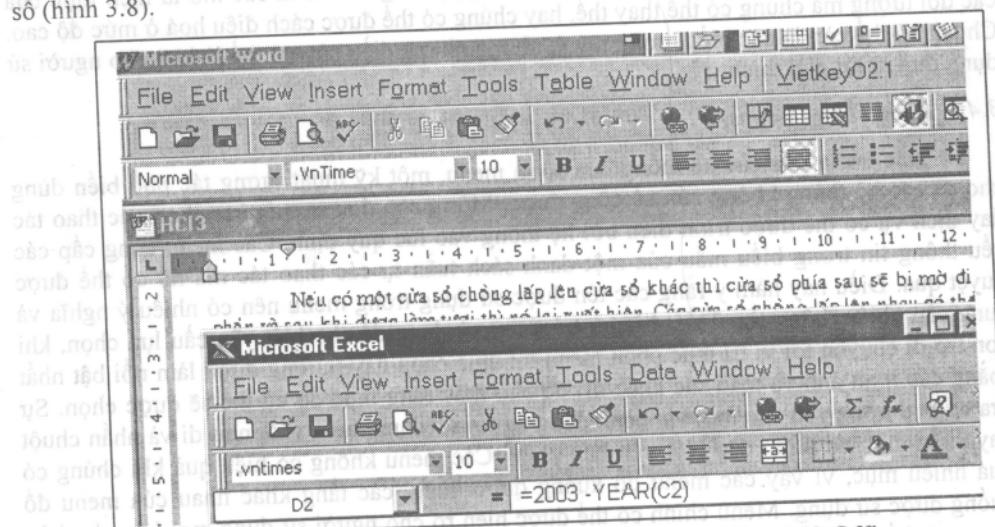
Stt	Họ và tên	Ngày sinh	Tuổi
1	Nguyễn Long Vương	01/01/1963	38
2	Trần Minh Hiếu	02/10/1975	28
3	Vũ Quán Hoà	03/02/1966	37
4	Phí Minh Khang	04/01/1975	28
5	Hồ Cảnh Hải	01/09/1967	36
6	Thành Không Đẹp	06/01/1981	22

**Hình 3.7. Xây dựng công thức trong bảng tính.**

sang dòng công việc khác.

Nếu có một cửa sổ chồng lấp lên cửa sổ khác thì cửa sổ phía sau sẽ bị mờ đi một phần và sau khi được làm rõ thì nó lại xuất hiện. Các cửa sổ chồng lấp lên nhau có thể gây ra các vấn đề như làm mờ đi các thông tin quan trọng, vì vậy các cửa sổ cũng được giữ bí mật khi chúng tiếp giáp nhưng không chồng lên nhau. Các cửa sổ có thể được đặt theo kiểu tầng, tức là mỗi cửa sổ mới được đặt hơi dịch sang trái và phía dưới cửa sổ trước. Trong một vài hệ thống cách bố trí này được cố định, trong các hệ thống khác nó có thể được lựa chọn bởi người sử dụng.

Thông thường các cửa sổ có kết hợp những thành phần khác với chúng nhằm gia tăng sự hữu ích, ví dụ các thanh cuộn được đính kèm vào cửa sổ để tăng sự linh hoạt, cho phép người sử dụng di chuyển nội dung cửa sổ lên hoặc xuống, từ cạnh này sang cạnh kia. Điều này khiến cửa sổ linh hoạt hơn và được xem như là một cửa sổ thực dùng cho một lượng lớn người quan sát thông tin bằng cách thao tác thanh cuộn. Trên cửa sổ thường có một thanh tiêu đề đính kèm phía trên đỉnh, nó dùng để nhận dạng cho người sử dụng, và có thể có các hộp đặc biệt ở góc của các cửa sổ để giúp chỉnh kích cỡ, đóng hoặc phóng to cửa sổ (hình 3.8).



**Hình 3.8. Các thành phần cửa sổ phần mềm văn phòng MS.Office.**

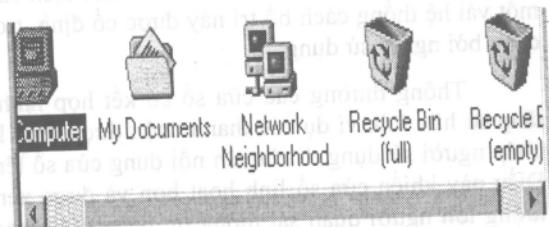
Trong một vài hệ thống lại cho phép các cửa sổ nằm bên trong các cửa sổ khác. Ví dụ, Microsoft Windows 3.1, mỗi ứng dụng có cửa sổ riêng của nó và trong phạm vi mỗi tài liệu lại có một cửa sổ. Nó có thể có các cách bố trí khác nhau trong các cửa sổ ứng dụng khác nhau.

#### **3.4.2. Biểu tượng (Icons)**

Các cửa sổ có thể được đóng lại và mất luôn hoặc chúng có thể được làm co lại cho các trình bày rút gọn. Một bức tranh nhỏ được sử dụng để miêu tả một cửa sổ đóng và cách trình bày này được biết như một biểu tượng. Với việc xuất hiện các biểu tượng thì nhiều cửa

sở có thể xuất hiện trên màn hình cùng lúc, sẵn sàng để mở rộng tới kích thước lớn nhất bằng việc nhấn vào biểu tượng. Việc làm co một cửa sổ thành biểu tượng được hiểu như là cửa sổ biểu tượng. Khi người sử dụng tạm thời không muốn tiếp tục dòng hội thoại, anh ta có thể hoãn hội thoại đó bằng biểu tượng của cửa sổ chứa hội thoại. Biểu tượng lưu giữ không gian trên màn hình của cửa sổ và dùng làm cách nhắc người sử dụng có thể phục hồi lại hội thoại sau đó bằng cách mở cửa sổ từ biểu tượng.

Hình 3.9 đưa ra một vài ví dụ về một số biểu tượng được sử dụng trong hệ thống cửa sổ điện hình. Các biểu tượng cũng có thể được sử dụng để mô tả những diện mạo khác của hệ thống, như sọt rác dùng để bỏ những tệp không cần thiết đến.

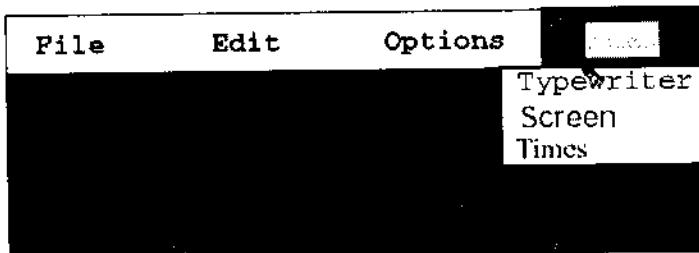


**Hình 3.9. Một số biểu tượng điển hình.**

Các biểu tượng có thể tạo nhiều biểu mẫu: chúng có thể là các mô tả hiện thực của các đối tượng mà chúng có thể thay thế, hay chúng có thể được cách điệu hoá ở mức độ cao. Chúng có thể còn là các biểu tượng tuỳ ý, nhưng những điều này có thể là khó cho người sử dụng đưa ra lời giải thích.

#### 3.4.3. Bảng chọn (Menus)

Đặc trưng của các hệ thống cửa sổ là menu, một kỹ thuật tương tác phổ biến dùng cho cả các hệ thống không cửa sổ cũng được. Một menu đưa ra một lựa chọn các thao tác hay dịch vụ có thể được trình diễn bởi hệ thống vào lúc quy định. Các menu cung cấp các kêu thông tin trong biểu mẫu của một danh sách tuần tự các thao tác mà nó có thể được duyệt qua. Điều này hàm ý rằng các tên được sử dụng trong menu nên có nhiều ý nghĩa và cung cấp nhiều thông tin. Thiết bị con trỏ được sử dụng để cho biết yêu cầu lựa chọn, khi con trỏ di chuyển tới vị trí mục chọn menu thì mục chọn này thường được làm nổi bật nhất (bằng các hình ảnh và màu sắc khác đi) để cho thấy rằng mục đó có thể sẽ được chọn. Sự lựa chọn thường yêu cầu một vài hành động người sử dụng thêm vào, như di và nhấn chuột hay nhấn một vài phím đặc biệt trên bàn phím. Các menu không có hiệu quả khi chúng có quá nhiều mục, vì vậy các menu đồ xuống được dùng, các tầng khác nhau của menu đồ xuống được sử dụng. Menu chính có thể được hiện rõ cho người sử dụng mọi lúc, như hộp menu có thể được kéo xuống hoặc ngang ra từ menu chính (hình 3.10). Các hộp menu được đặt trên đỉnh của màn hình (ví dụ MacOS) hay trên đỉnh mỗi cửa sổ (ví dụ Windows 3.1 hay Motif). Tuy nhiên, những lựa chọn hộp menu đọc theo cạnh của màn hình, hoặc đặt ngay ở giữa các cửa sổ trong vùng nền màn hình chính. Trong trường hợp nào đó menu chính có thể bị ẩn và khi có yêu cầu nó sẽ xuất hiện bất ngờ trên màn hình. Các menu pop-up thường được sử dụng để đưa ra các lựa chọn theo ngữ cảnh, ví dụ cho phép xem xét thuộc tính của các đối tượng đặc biệt trên màn hình. Tuy nhiên, trong một vài hệ thống chúng cũng được sử dụng để truy nhập các hành động toàn cục khi chuột nhấn vào nền màn hình. Các trình đơn đồ xuống được kéo xuống từ tiêu đề ở đỉnh của màn hình bằng việc di chuyển con trỏ chuột vào trong hộp tiêu đề và nhấn chuột. Các menu sụp xuống thì tương tự, ngoại trừ menu xuất hiện tự động khi con trỏ chuột di chuyển vào vùng tiêu đề, nếu không thì người



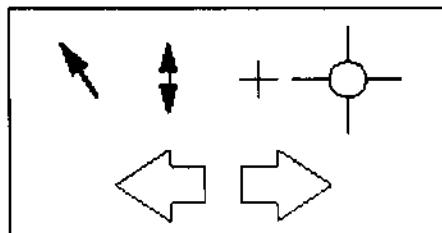
Hình 3.10. Kiểu menu bạch kim điển hình của dòng MacOS.

sử dụng phải nhấn chuột. Một vài menu là các menu kẹp, bởi vì chúng có thể được “ghim chặt” vào màn hình, luôn ở vị trí đó cho tới khi được yêu cầu rời ra. Các menu đồ xuống xuất hiện khi một vùng đặc biệt của màn hình được lựa chọn bởi một biểu tượng, nhưng chúng chỉ dừng với điều kiện là chuột được nhấn. Các lối vào khác tới menu lựa chọn được sắp xếp trong khung menu lựa chọn, con trỏ xuất hiện ở trung tâm của khung menu lựa chọn vì vậy khoảng cách di chuyển tới bất kỳ lựa chọn nào đều như nhau, điều này có những thuận lợi đó là dễ dàng lựa chọn các mục, thời gian lựa chọn cho mỗi mục là như nhau. So sánh điều này với một menu mẫu chúng ta thấy để lựa chọn các mục gần cuối của menu sẽ lâu hơn các mục đầu. Tuy nhiên, các lựa chọn dạng vòng tròn này đưa lên không gian màn hình nhiều hơn và ít phô biến hơn trong các giao diện. Các vấn đề lớn với các menu nói chung là gồm có những mục gi và cách nhóm các mục đó. Việc gộp quá nhiều mục tạo các menu quá dài hay quá nhiều mục, nhưng ngược lại việc nhóm sẽ gây ra những vấn đề trong các mục đó gắn liền với các đề tài cần đặt dưới tiêu đề, nhiều mục chọn cũng có thể được nhóm lại nhiều hơn dưới một tiêu đề. Trong các menu đồ xuống, nhấn menu nên được chọn để phản ánh chức năng của các mục chọn và các mục được nhóm trong các trong các menu bởi chức năng. Những nhóm này nên nhất quán xuyên suốt các ứng dụng. Các mục chọn menu nên được sắp đặt tuỳ theo tầm quan trọng và trình tự sử dụng, ngược lại về mặt chức năng (như ‘save’ và ‘delete’) nên để riêng để ngăn ngừa những lựa chọn bất ngờ không mong muốn với những hậu quả tai hại tiềm tàng.

#### 3.4.4. Con trỏ (Pointers)

Con trỏ là một phần quan trọng vì WIMP dùng chủ yếu định vị và lựa chọn. Dạng tương tác được yêu cầu bởi WIMP dựa rất nhiều vào việc trỏ và lựa chọn kiểu như các biểu tượng. Chuột cung cấp một thiết đầu vào có khả năng làm những việc như vậy, mặc dù các cần điều khiển (joysticks) và bóng xoay (trackballs) cũng là những sự lựa chọn. Người sử dụng được trình diễn với con trỏ trên màn hình được điều khiển bởi thiết bị đầu vào. Có rất nhiều loại hình dạng con trỏ cho ở hình 3.11. Các dạng con trỏ khác nhau thường được dùng để phân biệt các chế độ, ví dụ con trỏ bình thường có thể là một mũi tên, nhưng sẽ thay đổi thành mũi tên hai đầu khi vẽ một đường thẳng. Các con trỏ cũng được sử dụng để báo cho người sử dụng biết về sự hoạt động của hệ thống, ví dụ một con trỏ đồng hồ cát xuất hiện khi hệ thống đang bận đọc một tệp. Các dạng con trỏ là các biểu tượng tương tự nhau, chúng là các ảnh bitmap nhỏ, nhưng tất cả các con trỏ có một điểm nóng chính là vị trí mà chúng trỏ vào. Thỉnh thoảng điểm nóng là không rõ ràng từ sự xuất hiện của con trỏ, như

khi người sử dụng thấy khó kích vào những điểm đích nhỏ. Vì vậy khi thiết kế các con trỏ riêng thì phải chắc chắn rằng nó có một điểm nóng rõ ràng.



**Hình 3.11.** Một số dạng con trỏ.

Ngoài các thành phần như đã liệt kê, giao tiếp kiểu WIMP còn sử dụng thêm một số thành phần khác như phím lệnh (Command Button), Thanh công cụ (ToolBar), bảng màu và các hộp đối thoại.

### **3.5. Điều khiển trực tiếp**

Điều khiển trực tiếp là một dạng thống trị trong thiết kế các hệ thống văn phòng. Đó là kiểu giao tiếp WYSIWYG (What You See Is What You Get). Người dùng có thể giao tiếp thông qua bắt cứ biểu tượng có sẵn nào trên màn hình. Thí dụ để di chuyển một đối tượng tệp hay thư mục (folder) từ một thư mục này sang một thư mục khác, người dùng có thể sử dụng hình thức “gấp và thả”. Cũng tương tự như vậy, để xoá đi một đối tượng , người dùng có thể chọn rồi nhấn phím “Delete” hay gấp thả vào thùng rác. thậm chí muốn tráo đổi hai dòng trong một bảng tính, người dùng cũng có thể sử dụng . Giao tiếp kiểu này đơn giản, tự nhiên và dễ dàng cho hầu hết các đối tượng người dùng. Tuy nhiên, cũng cần chú ý tránh sự nhầm lẫn khi gấp thả: thí dụ người dùng chỉ cần sao (copy) chứ không cần chuyển (move).

Cũng do bản chất này mà giao tiếp kiểu điều khiển trực tiếp hầu như không được áp dụng trong thiết kế công nghiệp.

### **3.6. Ngữ cảnh tương tác**

Chúng ta đang quan tâm đến tương tác giữa một người sử dụng và một hệ thống và điều này bị ảnh hưởng bởi việc thiết kế giao diện như thế nào. Chúng ta hầu như đã chú ý đến một vài nhân tố vật lý trong môi trường mà nó có thể trực tiếp làm ảnh hưởng đến chất lượng của tương tác. Đây là phần ngữ cảnh ở vị trí mà tương tác xác định. Tuy nhiên điều này có vẻ như một hoạt động đơn người sử dụng, mặc dù phức tạp, máy móc. Trong thực tế, những người sử dụng làm việc trong một phạm vi có tính chất xã hội rộng rãi và phạm vi một tổ chức. Điều này cung cấp ngữ cảnh rộng hơn cho tương tác và có thể tác động đến phạm vi hoạt động và động cơ thúc đẩy của người sử dụng. Trong phần này chúng ta sẽ hạn chế việc thảo luận về những nhân tố của tổ chức và xã hội có thể có trên tương tác của người sử dụng với hệ thống. Những điều này có thể không phải là những nhân tố vượt quá điều mà người thiết kế quy định. Tuy nhiên điều quan trọng là biết được những tác động như vậy để hiểu được người sử dụng và phạm vi đầy đủ của công việc.

### Chương 3: Các mô hình và các dạng tương tác người máy

Sự xuất hiện của người khác trong môi trường làm việc sẽ làm ảnh hưởng đến hiệu suất của người làm việc trong bất kỳ công việc nào. Trong trường hợp những người đồng nghiệp thì sự cạnh tranh sẽ gia tăng hiệu suất, chí ít là những công việc đã biết. Tương tự, yêu cầu đưa vào quản lý và chất lượng cao đã cải thiện hiệu suất trên các công việc này. Tuy nhiên, khi nó đi đến giành được những kỹ năng mới thì sự xuất hiện của các nhóm này có thể hạn chế hiệu suất, do việc lo sợ sự thất bại. Do vậy bí mật là quan trọng cho phép người sử dụng cơ hội thử nghiệm.

Để mà thực hiện tốt, người sử dụng phải được thúc đẩy. Có một số nguồn có thể thúc đẩy, cũng như những điều đó chúng ta hầu như đã thúc đẩy, kể cả sự e ngại, lòng trung thành, tham vọng và sự tự mãn. Điều cuối cùng là ảnh hưởng bởi sự hiểu biết của người sử dụng về chất lượng công việc được làm, nó dẫn đến sự thoả mãn công việc. Nếu một hệ thống gây khó khăn cho người sử dụng thực hiện những công việc cần thiết hay gây ra sự khó chịu khi sử dụng, sự thoả mãn công việc của người sử dụng và trình tự thực hiện sẽ giảm bớt.

Người sử dụng sẽ mất đi động cơ thúc đẩy nếu một hệ thống được giới thiệu là không đổi chọi được với yêu cầu thực tế của công việc được thực hiện. Thường các hệ thống được lựa chọn và giới thiệu bởi người quản lý hơn là chính bản thân những người sử dụng. Trong một vài trường hợp thì sự am hiểu của người quản lý về công việc có thể dựa trên việc quan sát kết quả chứ không phải là các hoạt động thực tế. Vì vậy hệ thống đã giới thiệu có thể làm đảo lộn hướng công việc, điều đó là không tốt cho người sử dụng. Nếu điều này xảy ra có thể có ba kết quả: hệ thống sẽ bị loại bỏ, người sử dụng sẽ thấy bức bối và không có mục đích, hoặc người sử dụng sẽ sửa lại tương tác có ý định cho các yêu cầu của riêng anh ta. Điều này cho biết sự quan trọng phải kéo theo những người sử dụng thực tế vào trong quá trình thiết kế.

Mặt khác, việc giới thiệu kỹ thuật mới có thể chứng tỏ là một sự thúc đẩy tới người sử dụng, đặc biệt nếu nó được thiết kế tốt, được tích hợp với công việc hiện tại của người sử dụng. Việc cung cấp sự phản hồi là một nguồn thúc đẩy quan trọng cho người sử dụng. Nếu không nhận được sự phản hồi trong một phiên thì người sử dụng có thể chán nản, không có mục đích hoặc lo lắng, không tin tưởng các hành động được thực hiện có thành công hay không. Nói chung, một hành động nên có một hiệu quả rõ ràng để ngăn cản sự lộn xộn này và cho phép dễ dàng phục hồi trong trường hợp bị lỗi. Tương tự nếu hệ thống trễ xảy ra thì sự phản hồi có thể được sử dụng để ngăn cản thất bại cho người sử dụng – người sử dụng biết là cái gì đang xảy ra và không ngạc nhiên nếu hệ thống vẫn đang làm việc.

# 4

## THIẾT KẾ TƯƠNG TÁC NGƯỜI - MÁY

### Nội dung

- 4.1. Tổng quan
- 4.2. Các mô thức cho tính dùng được
- 4.3. Các nguyên tắc cho tính dùng được
- 4.4. Thiết kế giao tiếp người dùng - máy tính
- 4.5. Thiết kế lập và mẫu thử

### 4.1. Tổng quan

Thiết kế đảm bảo tính dùng được (Usability) tối đa là mục tiêu của việc thiết kế các hệ thống tương tác. Tính dùng được có thể hiểu một cách đơn giản ở đây là hệ thống phải cho phép người dùng thực hiện các nhiệm vụ một cách hiệu quả như đã chỉ trong phần 0.1.2 (chương mở đầu) theo chuẩn ISO 9241-11. Việc thiết kế các hệ thống tương tác cần trả lời hai câu hỏi:

- 1) Một hệ thống tương tác được phát triển có đảm bảo tính dùng được?
- 2) Liệu tính dùng được của hệ thống tương tác có thể chứng minh hay lượng hoá được không?

Có hai cách tiếp cận nhằm trả lời hai câu hỏi trên. Cách tiếp cận thứ nhất là xem xét các hệ thống tương tác được thiết kế thành công và làm mô thức cho việc phát triển các hệ thống tương tác trong tương lai. Cách tiếp cận thứ hai mang tính lý thuyết hơn, nó nhằm đề xuất các nguyên tắc trừu tượng cho việc tương tác hiệu quả từ các tri thức về tâm lý học, lý thuyết tính toán và xã hội học trong các lĩnh vực liên quan. Các nguyên tắc này sẽ điều khiển thiết kế và đánh giá một sản phẩm theo tập các tiêu chí đặt ra.

Các mô thức cung cấp cho chúng ta niềm tin là các hệ thống ngày nay có tính dùng được cao hơn các hệ thống trong quá khứ và các hệ thống trong tương lai sẽ tốt hơn các hệ thống hiện tại.

Công nghệ hiện đại ngày càng tạo ra các máy tính mạnh hơn, song công nghệ một mình không đủ để làm tăng tính dùng được. Chìa khoá quan trọng đó là tính sáng tạo và các công nghệ làm tăng sức mạnh của con người. Mô thức thì phụ thuộc vào công nghệ và bản chất các ứng dụng, còn nguyên tắc thì độc lập với công nghệ và chỉ phụ thuộc vào mức độ hiểu con người - thành phần quan trọng trong tương tác. Trong các phần dưới đây, chúng ta sẽ đề cập chi tiết đến hai cách tiếp cận cũng như các chuẩn trong thiết kế.

### 4.2. Các mô thức cho tính dùng được

Chúng ta sẽ xem xét trong phần này một số mô thức chính trong số 14 mô thức khác nhau [2-4.2]. Nó không loại trừ lẫn nhau mà có sự bổ sung cho nhau. Chúng ta không đơn thuần là xem xét lịch sử phát triển của các hệ thống tương tác mà nhấn mạnh vào các đổi mới của tương tác, vào tính ưu việt của mỗi phương thức.

#### 4.2.1. Mô thức thời gian phân chia (time sharing)

Một trong những đóng góp quan trọng này sinh từ các nghiên cứu tiên tiến đó là kỹ thuật chia thời gian. Nhờ kỹ thuật này mà một máy tính có thể hỗ trợ đa người dùng. Trước đây, con người hay nói đúng hơn là các lập trình viên chỉ thực hiện theo lô (batch) theo cách một tập lệnh có thể lưu trên bìa hay băng giấy được nạp vào máy tính và các thao tác viên có thể thực hiện chúng một cách riêng lẻ. Kỹ thuật này bắt đầu có từ những năm 1960 và tạo cho lập trình một phong cách tương tác thực sự. Máy tính cũng trở thành một thành viên trong đối thoại với mỗi người dùng và sự gia tăng thông lượng thông tin giữa người dùng và máy tính cho phép con người ngày càng trở nên tích cực và hợp tác kịp thời. Với kỹ thuật này, tương tác người dùng - máy tính đã trở nên có khả năng.

## **Chương 4: Thiết kế giao tiếp người - máy**

---

### **4.2.2. Các thiết bị hiển thị quan sát (Video Display Units - VDU)**

Vào đầu những năm 1950, các nhà nghiên cứu đã thực nghiệm khả năng biểu diễn và điều khiển thông tin của một máy tính trong dạng ảnh và trên một thiết bị VDU. Các màn hình hiển thị này cung cấp nhiều tính năng quan trọng hơn giấy trong cách biểu diễn thông tin. Lúc đầu các thiết bị này được sử dụng trong quân sự.

Các nghiên cứu tiếp theo của Sketchpad cho phép các thao tác viên dùng máy tính để tạo một cách nhanh chóng các mô hình trực quan, tinh vi trên một màn hình hiển thị, giống như màn hình tivi. Các mẫu quan sát có thể lưu trong bộ nhớ của máy tính giống như các dữ liệu khác và có thể được xử lý bởi bộ xử lý. Sketchpad minh họa hai ý tưởng quan trọng. Ý tưởng thứ nhất là máy tính có thể được sử dụng không chỉ đơn thuần là xử lý dữ liệu. Nó có thể mở rộng khả năng của người dùng nhằm trùu tượng hoá cách thức với một vài mức độ chi tiết khác nhau: trực quan hoá và điều khiển nhiều cách biểu diễn khác nhau từ chính một thông tin. Việc trùu tượng hoá này không chỉ giới hạn trong cách biểu diễn dữ liệu bằng một chuỗi bit bên trong bộ nhớ máy tính mà còn làm cho việc biểu diễn thực sự trực quan. Máy tính được chế tạo để nói ngôn ngữ người dùng thay vì người dùng bị cưỡng bức nói ngôn ngữ của máy tính. Ý tưởng thứ hai là sự đóng góp quan trọng của các ý tưởng sáng tạo trong toàn bộ lịch sử máy tính.

### **4.2.3. Các công cụ lập trình (Programming Toolkits)**

Từ những năm 1950, Douglas Engelbart đã có ham vọng dùng công nghệ máy tính như là một phương tiện cho hoạt động giải quyết bài toán của con người - dùng máy tính để dạy học. Bằng cách làm tăng tri thức của con người, hy vọng sẽ làm tăng khả năng của con người để tiếp cận các tình huống phức tạp, hiểu và thích ứng với các nhu cầu riêng và dẫn đến lời giải cho bài toán.

Các bí mật nhằm cung cấp các thiết bị tính toán trợ giúp khả năng giải quyết vấn đề của con người là cung cấp các ý tưởng đúng. Ý tưởng của việc xây dựng các thành phần của hệ thống tính toán, cái sẽ cho phép chúng ta xây dựng lại một hệ thống phức tạp hơn được gọi là quá trình khởi động (Boostapping) và đã từng được sử dụng mở rộng trong mọi tính toán. Sức mạnh của các công cụ lập trình ở chỗ là thành phần nhỏ, được hiểu một cách thấu đáo là có thể tổ hợp theo một cách thức cố định nhằm tạo ra các công cụ lớn hơn. Một khi bộ công cụ lớn hơn được hiểu rõ, chúng có thể tiếp tục được tổ hợp để tạo nên bộ công cụ lớn hơn nữa và quá trình cứ tiếp tục.

### **4.2.4. Máy tính cá nhân**

Các công cụ lập trình cung cấp một phương tiện cho những gì cùng với cách thức tính toán quan trọng nhằm tăng năng suất. Từ những thập niên 70, chúng ta đã thấy xuất hiện các máy tính mạnh nhằm xử lý lượng lớn, tính toán văn bản hay không. Seymour Papert đã phát minh ra ngôn ngữ lập trình đồ họa LOGO với mục đích dạy trẻ em có thể học về các đối tượng hình học đơn giản như các hình chữ nhật, hình tròn với cơ chế “con rùa”. Bằng cách thích ứng hoá ngôn ngữ lập trình đồ họa cho một mô hình mà trẻ em có thể hiểu và sử dụng, Papert đã chứng minh một châm ngôn có giá trị cho phát triển các hệ thống tương tác - không có gì mạnh hơn một hệ thống nếu nó luôn dễ sử dụng.

Alan Kay, người chịu ảnh hưởng của cả Engelbart và Papert đã nhận ra rằng sức mạnh của hệ thống là nó cho phép truy nhập bởi các người dùng mới như hệ thống LOGO. Đầu những năm 1970, quan điểm của ông ta về tương lai của máy tính là chúng phải nhỏ, mạnh và thiết kế cho một người dùng - các máy tính cá nhân. Cùng với sự phát triển của công nghệ, hiện tại khó phân biệt được những gì cấu thành nên các máy tính cá nhân, các trạm làm việc hay các máy tính lớn.

### 4.2.5. Hệ thống Window và giao diện WIMP

Với sự thành công của các máy tính cá nhân, công nghệ máy tính tập trung vào các đòi hỏi của đơn người dùng trong quá trình đối thoại với máy tính nhằm thực hiện một số công việc. Con người có khả năng suy nghĩ về nhiều vấn đề tại một thời điểm và trong khi thực hiện một số phần của công việc này, họ thường cắt dòng suy nghĩ hiện thời để theo đuổi một số phần khác của công việc. Một hệ thống máy tính cá nhân thường khuyến khích người dùng tiến lên nhằm xuyên suốt mọi nhiệm vụ cần thiết để hoàn thành một số mục tiêu, từ khi bắt đầu cho đến khi kết thúc không bị chêch hướng không tương ứng với các dạng công việc chuẩn. Nếu máy tính cá nhân có thể là một đối tác đối thoại hiệu quả, nó phải là mềm dẻo trong khả năng thay đổi các chủ đề như con người.

Tuy nhiên, khả năng gửi các nhu cầu tới một nhiệm vụ người dùng không phải là một đòi hỏi. Hệ thống máy tính phải phản ứng lại các kích thích đến từ người dùng, do vậy nó phải chịu trách nhiệm về các đối thoại ngẫu hứng của người dùng. Khi người dùng khởi thảo nhiều hơn một dự định hoạt động trong một khoảng thời gian, thì rất khó duy trì trạng thái của các luồng hoạt động phủ nhau. Do vậy, rất cần thiết để đối tác đối thoại máy tính cảnh báo (ngăn chặn) ngữ cảnh của mỗi luồng đối thoại sao cho người dùng có thể phân biệt được chúng.

Một cơ chế biểu diễn để thực hiện phân chia đối thoại là tách biệt một cách vật lý việc biểu diễn của nhiều luồng lô gic khác nhau của việc đối thoại người dùng-máy tính trên thiết bị hiển thị. Window là cơ chế chung nhất liên kết với các thiết bị vật lý và không gian hiển thị tách biệt lô gic.

Càng ngày người dùng máy tính càng trở nên quen thuộc với tương tác dựa vào window, icon, menu và con trỏ mà chúng ta gọi là giao diện WIMP.

### 4.2.6. Cảnh trí (Metaphor)

Trong việc phát triển ngôn ngữ LOGO để dạy vẽ cho trẻ em, Papert đã dùng cảnh trí con rùa và trẻ em dễ dàng xác định hiện tượng trong thế giới thực và nhanh chóng hiểu được cách thức tạo nên các bức tranh. Cảnh trí được sử dụng khá thành công để dạy các khái niệm mới theo nghĩa khi đã biết một cái có thể suy ra các cái tương tự. Chúng ta đã thấy cách cảnh trí được sử dụng để miêu tả hoạt động của nhiều đối tượng giao tiếp như: cửa sổ, thực đơn, phím lệnh và bảng màu. Xerox và Star là các trạm làm việc đầu tiên dùng cảnh trí cho các ứng dụng văn phòng.

Nhiều tranh luận đã diễn ra về việc thế nào là cảnh trí tốt để làm tăng tính tham gia giữa người dùng và các ứng dụng máy tính. Cái nguy hiểm của cảnh trí là nhiều khi cảnh trí như nhau song hành vì lại khá khác nhau và dễ bị sai. Thí dụ như bàn phím máy tính giống

như máy chữ chuẩn và phím gõ dấu cách đối với máy chữ không có nghĩa gì, song với máy tính thì lại rất có ý nghĩa. Nó tạo nên một ký tự như các ký tự khác và người quen đánh máy chữ thì khó lòng suy đoán. Một vấn đề tương tự cũng này sinh với phần lớn cảnh trí. Dù rằng cảnh trí văn phòng hấp dẫn một cách tự nhiên, song lại có một chút vấn đề khi làm việc với máy tính, bởi vì thực sự không có một văn phòng thực tế nào tương đương với một số thao tác với văn phòng máy tính.

Một vấn đề khác với cảnh trí đó là khuynh hướng văn hóa thiên về chân dung. Với sự tăng trưởng có tính quốc tế hóa phần mềm, khó đảm bảo rằng một cảnh trí sẽ được áp dụng một cách xuyên biên giới. Còn nhiều vấn đề này sinh nữa nhất là với hiện tại ảo và hậu quả là người dùng thường bị “lắp ráp” với các thiết bị vẽ đặc biệt và như vậy hệ thống phải định vị chúng và phải giải thích các cử động của họ một cách chính xác.

### 4.2.7. Điều khiển trực tiếp (Direct manipulation)

Vào đầu những năm 1980 khi mà giá thành của phần cứng đồ họa chất lượng cao tiếp tục giảm, các nhà thiết kế bắt đầu thấy rằng các sản phẩm của mình đã chiếm đa số khi mà nội dung trực quan tăng lên. Chừng nào đôi thoại hệ thống-người dùng còn được duy trì theo kiểu một hướng: từ lệnh người dùng đến lời nhắc dòng lệnh máy tính thì việc tính toán còn ít phổ biến và chỉ dừng trong số ít các hắc cơ. Sự trực quan và phản hồi nhanh chóng trên thiết bị với độ phân giải cao hoặc qua hệ thống âm thanh sẽ tạo nên khả năng cung cấp thông tin đánh giá cho các hành động của người dùng.

Sự phản hồi nhanh chóng (rapid feedback) đã trở nên một đặc trưng quan trọng của kỹ thuật tương tác, được gọi là *điều khiển trực tiếp*. B. Shneiderman [19] đã nhấn mạnh các đặc trưng quan trọng của giao diện điều khiển trực tiếp như sau:

1. Tính nhìn thấy được của các đối tượng quan tâm.
2. Gia tăng hoạt động trên giao diện với phản hồi nhanh chóng của mọi hành động.
3. Tính thuận nghịch của mọi hành động và do vậy người dùng được khuyến khích khám phá không bị các hậu quả nặng nề.
4. Mọi hành động có cú pháp chính xác, do đó mỗi hành động của người dùng đều là các thao tác hợp lệ.
5. Thay thế ngôn ngữ dòng lệnh phức tạp bởi các hành động nhằm điều khiển trực tiếp các đối tượng thấy được.

Sự thành công của thương trường đã minh chứng tính dùng được cổ hữu của giao tiếp kiểu điều khiển trực tiếp: giao diện phổ biến cho các máy tính cá nhân như Macintosh của hãng Apple. Giao diện điều khiển trực tiếp cho cảnh trí văn phòng đòi hỏi các tài liệu và các thư liệu (folder) phải trực quan với người dùng như là các biểu tượng biểu diễn tệp và thư mục. Một thao tác như chuyển một tệp từ thư mục này đến thư mục khác được thể hiện như một hành động trên một tài liệu nhìn thấy được bằng cách “gấp và thả” trên văn phòng từ một thư liệu này sang một thư liệu bên cạnh. Trong bài báo của mình về “thiết kế lấy người dùng làm trung tâm”, D. Norman và Draper đã viết:

## **Chương 4: Thiết kế giao tiếp người - máy**

“Trong một hệ thống được xây dựng trên cảnh trí mô hình- thế giới, giao diện tự nó là một thế giới, ở đó người dùng có thể hành động và hệ thống thay đổi trạng thái đáp ứng hành động này. Thế giới của giao tiếp được biểu diễn một cách tường minh và không có gì trung gian giữa người dùng và thế giới. Việc sử dụng chính xác cảnh trí mô hình-thế giới có thể tạo nên cảm giác trong hành động của người dùng trên các đối tượng trong chính lĩnh vực nhiệm vụ đó. Chúng ta gọi sắc thái này của tính trực tiếp là sự cam kết trực tiếp”.

Trong cảnh trí mô hình - thế giới, vai trò của giao tiếp không khác gì môi trường trung gian giữa người dùng và hệ thống cơ sở. Một hệ quả của mô thức điều khiển trực tiếp là không có sự phân biệt rõ ràng giữa đầu vào và đầu ra. Trong mô hình khung tương tác nói trong chương 3, chúng ta đã đề cập tới sự khớp nối việc biểu hiện đầu vào của một người dùng bằng một vài ngôn ngữ đầu ra. Trong một hệ thống điều khiển trực tiếp, việc biểu hiện đầu ra được dùng để hình thành một chuỗi các biểu hiện đầu vào. Biểu tượng của tài liệu là một sự biểu hiện của đầu vào trong cảnh trí văn phòng, nhưng biểu tượng này được người dùng dùng để khớp các thao tác chuyên. Sự kết hợp của đầu vào và đầu ra này được phản ánh trong các công cụ lập trình khi mà các đối tượng không được xem như các đối tượng vào ra loại trừ nhau. Đúng hơn, các đối tượng bao bọc cả hai ngôn ngữ vào và ra, và chúng ta coi chúng là các *đối tượng tương tác*.

Kiểu tương tác này phản ánh tính trực quan và được biết đến với tên “WYSIWYG”- cái nhìn thấy là cái có được. Cái nhìn thấy trên màn hình hiển thị, thí dụ khi bạn đang sử dụng một hệ soạn thảo văn bản, không phải là tài liệu hiện thời bạn sẽ tạo ra lúc kết thúc. Đúng hơn, đó chỉ là sự biểu diễn hay sự làm đẹp của văn bản mà chúng ta sẽ thấy khi kết thúc soạn thảo. Ngụ ý của một giao tiếp WYSIWYG là ở chỗ sự khác nhau giữa biểu diễn và sản phẩm cuối cùng là rất nhỏ, và người dùng dễ dàng quan sát sản phẩm cuối cùng từ sự biểu diễn của máy tính. Do vậy, trong thí dụ soạn thảo văn bản ta có thể hình dung toàn bộ dáng vẻ của văn bản sẽ như thế nào qua ảnh của nó trên màn hình, giảm thiểu một vài ước đoán khi định dạng bản in cuối cùng.

Giao tiếp WYSIWYG rõ ràng là đơn giản và tức thì để ánh xạ giữa việc biểu diễn và sản phẩm cuối cùng, đó là bản chất của giao tiếp này. Trong thuật ngữ của khung tương tác, việc quan sát sự biểu hiện đầu ra được thực hiện một cách đơn giản sao cho việc khẳng định sự thực hiện mục đích là trực tiếp. Tuy nhiên, giao tiếp WYSIWYG không phải là phương thuốc hữu hiệu cho tính dùng được. Thí dụ như trang văn bản chúng ta thấy trên màn hình không hoàn giống như trang được in ra.

### **4.2.8. Ngôn ngữ ngược với hành động**

Giao tiếp kiểu điều khiển trực tiếp làm cho một số nhiệm vụ được thực hiện dễ và chính xác hơn, trong khi nó cũng làm cho một số nhiệm vụ khó khăn hơn, nếu không nói là không thể. Trái ngược với hiểu biết chung, thường không đúng khi cho là hành động nói nhiều hơn việc làm. Hình ảnh mà chúng ta dựng lên cho giao tiếp điều khiển trực tiếp như là sự thay thế một hệ thống cơ sở bằng một tập các quan tâm của người dùng. Hành động thực hiện trên giao tiếp thay thế nhu cầu hiểu ngữ nghĩa ở bất cứ mức nào của hệ thống. Một hình ảnh khác là coi giao tiếp như là người đối thoại hay phương tiện giữa người dùng và hệ thống. Người dùng cung cấp các mệnh lệnh giao tiếp và tiếp theo chịu trách nhiệm về giao thông.

## *Chương 4: Thiết kế giao tiếp người - máy*

---

tiếp để thấy được các mệnh lệnh này đã được thực hiện xong ra sao. Giao tiếp người dùng-máy tính được thực hiện bởi ngôn ngữ gián tiếp thay vì hành động trực tiếp.

Chúng ta có thể gán hai giải thích đầy ý nghĩa cho mô thức ngôn ngữ này. Cái thứ nhất bắt buộc người dùng phải hiểu các chức năng của hệ thống cơ sở và giao diện như là sự đối thoại không cần phải thực hiện nhiều chuyên dịch. Thực vậy, cách giải thích này của mô thức ngôn ngữ tương tự như kiểu tương tác đã tồn tại trước khi tương tác kiểu điều khiển trực tiếp ra đời.

Cách giải nghĩa thứ hai không bắt buộc người dùng phải hiểu cấu trúc của hệ thống cơ sở. Giao tiếp đóng nhiều hơn một vai trò tích cực như là nó phải thông dịch giữa thao tác mong muốn như là yêu cầu bởi người dùng và các thao tác của hệ thống có thể để đáp ứng yêu cầu. Vì nó luôn kích hoạt nên một số người coi giao diện như là một tác tử (agent). Chúng ta có thể thấy cách thức này của mô thức ngôn ngữ khi làm việc với một hệ thống trích rút thông tin.

### *4.2.9. Siêu văn bản (HyperText)*

Siêu văn bản là một khái niệm khá quen thuộc với chúng ta hiện nay, nhất là trong môi trường WEB. Nó vừa là một khái niệm vừa là một kỹ thuật cho phép truy nhập một cách ngẫu nhiên vào một văn bản. Khái niệm văn bản cũng được mở rộng, không phụ thuộc vào không gian lưu trữ vật lý, một phần của văn bản có thể nằm trên một tệp khác trên cùng máy, trên một máy khác trong mạng cục bộ, thậm chí trên một website trên Internet. Với người dùng đó là một văn bản lô gíc và duy nhất. Một điều khác nữa là các phần văn bản có thể sản sinh bởi các ứng dụng khác nhau.

Ý tưởng này đã được nghiên cứu bởi Vannever Bush từ năm 1945, đăng trên tạp chí Atlantic Monthly với tiêu đề “ As We May Think”. Để theo kịp với sự gia tăng của các luồng thông tin, ý tưởng này nhằm làm tăng khả năng của con người để lưu trữ và trích rút các mẫu tri thức được liên kết bằng cách bắt chước khả năng của chúng ta nhằm tạo các liên kết kết hợp ngẫu nhiên. Vào năm 1960, một sinh viên tên là Ted Nelson đã thử xây dựng một văn phòng với khả năng lưu trữ một số lượng lớn các ánh sao của các văn bản bằng cách tạo ra một ngôn ngữ máy để thực hiện ý tưởng này song đã thất bại. Tuy nhiên, ý tưởng này đã cung cấp một cách tiếp cận mới về văn bản, có thể xây dựng một hệ thống trích rút thông tin và chế biến dựa trên ý tưởng liên kết, không tuyến tính và nhiều dạng phương tiện.

Thường với văn bản truyền thống trên giấy, người đọc sẽ duyệt theo một trình tự tuyến tính - từ đầu đến cuối. Tuy nhiên, trong một văn bản thường vẫn chứa các ý tưởng như chủ thích, nó có thể lôi cuốn người đọc đi tìm những chủ đề khác. Cách thức truy nhập tuyến tính không hỗ trợ nhiều cho mong muốn truy nhập ngẫu nhiên và các nhiệm vụ truy quét liên kết.

Gần hai thập kỷ sau mơ ước của Nelson, thuật ngữ siêu văn bản lần đầu tiên đã xuất hiện trên thương trường. Nhằm phản ánh việc sử dụng không tuyến tính, dựa vào liên kết cho không chỉ lưu trữ, trích rút các thông tin văn bản, khái niệm siêu phương tiện (hypermedia) và đa phương tiện (multimedia) ra đời và được dùng trong lưu trữ không tuyến tính của mọi phương tiện điện tử.

### 4.2.10. *Đa thể thức(Multi-modality)*

Phần lớn các hệ thống tương tác sử dụng bàn phím kinh điển và có thể có một thiết bị trỏ như chuột để phục vụ cho đầu vào và có một thiết bị hiển thị như màn hình màu phục vụ cho đầu ra. Các thiết bị vào ra này có thể xem như các kênh truyền thông cho hệ thống và chúng tương ứng với một số kênh truyền thông của con người. Một hệ thống đa thể thức là một hệ thống sử dụng nhiều kênh truyền thông của con người. Mỗi một kênh khác nhau đối với người dùng là một thể thức tương tác. Theo nghĩa này, mọi hệ thống tương tác có thể xem như đa thể thức với người dùng luôn sử dụng kênh thị giác và xúc giác (haptic) trong việc điều khiển máy tính.

Các hệ thống đa thể thức hiện đại liên quan đến việc sử dụng đồng thời của nhiều kênh truyền thông cho cả đầu vào và đầu ra. Con người xử lý thông tin một cách khá tự nhiên bằng cách sử dụng đồng thời nhiều kênh truyền thông khác nhau. Các hệ thống đa thể thức, đa phương tiện và hiện thực ảo là những vấn đề đang được quan tâm trong thiết kế các hệ thống tương tác.

### 4.2.11. *WEB*

Theo nhiều chuyên gia và tự chúng ta cũng thấy WEB là một trong các phát triển có ý nghĩa nhất trong giao tiếp máy tính gần đây. Mọi trang Web đều có thể bao gồm văn bản, ảnh màu, phim, âm thanh và quan trọng nhất là các siêu liên kết để liên kết các trang khác nhau. Dự án xây dựng WEB đã được Tim Berners Lee đề xuất vào năm 1989 tại Trung tâm nghiên cứu Hạt nhân châu Âu (CERN) đặt tại Gionevơ, Thụy Sĩ với mục đích phân phối các dữ liệu khoa học từ CERN đến các trung tâm vật lý trên thế giới. Vào năm 1991, trình duyệt WEB hướng văn bản lần đầu tiên đã được thực hiện. Đầu năm 1993, một số trình duyệt WEB đồ họa, có ý nghĩa nhất như Mosaic đã được phát triển tại NSCA ở Champaign, bang Illinois Hoa Kỳ. Đó cũng là thời điểm xác định sự trưởng thành vượt bậc, sự gia tăng nhanh chóng chi phối các giao dịch trên Internet và làm thay đổi cách nhìn của công chúng về máy tính.

Dù bắt đầu xuất hiện ở những năm 1969, Internet vẫn chưa trở thành một mô thức trong tương tác, và chỉ đến khi có sự xuất hiện và tính dễ dùng của các giao tiếp đồ họa của các trình duyệt Web. Các trình duyệt này cho phép người dùng truy nhập các thông tin đa phương tiện một cách dễ dàng chỉ bằng một thiết bị định vị (chuột) và nhấn phím. Điều này thúc đẩy sự tích hợp tính toán và truyền thông trong suốt với người dùng; mọi cái nó thực hiện chính là cái nó có thể nhận được phiến bản thông tin hiện thời một cách tức thì. Hơn nữa, ngôn ngữ dùng để tạo ra các văn bản đa phương tiện là tương đối đơn giản, mở ra một cơ hội để xuất bản thông tin cho mọi lĩnh vực và liên kết với con người.

Ngày nay, Web là một trong những lý do chính liên kết người dùng với Internet, và nhanh chóng trở thành một hoạt động chính của con người trong cả công việc lẫn khi giải trí. Web là một hiện tượng xã hội, hơn mọi thứ khác, người dùng bị lôi cuốn vào các ý tưởng mà máy tính đã định sẵn và liên kết họ với những con người thú vị, với các địa danh hấp dẫn.

### 4.2.12. *Giao tiếp dựa vào tác tử (Agents based interface)*

## **Chương 4: Thiết kế giao tiếp người - máy**

---

Trong thế giới con người, tác từ là những con người đại diện cho một người nào đó, thí dụ các đại lý bán và mua hàng cho khách hàng, các đại lý sách thi tìm nhà xuất bản cho các tác giả, các hãng du lịch đặt khách sạn và vé cho các chuyến du lịch, ...

Tương tự như vậy, các tác từ phần mềm hoạt động nhân danh người dùng con người trong thế giới điện tử, thí dụ như tác từ thư tín lọc thư cho bạn, bộ duyệt Web đi dò tìm các trang Web/ hay các tài liệu trên các trang Web. Các tác từ có thể thực hiện các nhiệm vụ lặp , quan sát và trả lời các sự kiện khi người dùng vắng mặt, thậm chí có thể học từ các hành động của người dùng.

Một số tác từ chỉ có thể làm các công việc đơn giản kiểu luật *If then...*; tuy nhiên một số tác từ lại có thể có tính động và thông minh ở các mức độ khác nhau. Thí dụ, tác từ dùng trong khi tính toán công thức như tính tổng chặng hạn trong bảng tính Excel, có thể trợ giúp khi ta đặt công thức  $\Sigma$  vào ô: nó có thể điền tính theo hàng dọc hay hàng ngang phụ thuộc vào ngữ cảnh các ô phía trên hay phía trái nó.

Các mô thức đề cập ở trên, mỗi mô thức đều có thể mạnh riêng. Các hệ thống dựa vào dòng lệnh hoạt động như một phần tử trung gian: người dùng yêu cầu hệ thống thực hiện một điều gì đó, hệ thống thực hiện và có thể đáp ứng và hình thành trả lời lại cho người dùng. Ngược lại, giao tiếp kiểu điều khiển trực tiếp nhấn mạnh vào chính hành động người dùng và có thể tăng lên nhờ các công cụ trong thế giới điện tử. Tác từ hoạt động nhân danh người dùng, có thể không nhất thiết được mã hóa bằng kiểu ngôn ngữ. Tuy nhiên, không giống như các mô thức trung gian, kinh điển, một tác từ là một hành động tiêu biểu trong thế giới mà người dùng cũng có thể hành động trên đó.

### **4.3. Các nguyên tắc cho tính dùng đượ**

Các nguyên tắc cho tính dùng được như đã đề cập trong chương Một theo Nielson (1993), gồm 5 nguyên tắc chính:

- 1) Tính dễ học.
- 2) Tính hiệu quả.
- 3) Tính dễ nhớ.
- 4) Tính dự đoán lỗi.
- 5) Đáp ứng tính chủ quan.

Tuy nhiên trong phần này, để dễ trình bày, phát triển và chi tiết hóa các nguyên tắc, chúng ta tập trung vào ba tiêu chí chính [2]:

- Tính dễ học (learnability).
- Tính mềm dẻo (flexibility).
- Tính vững chắc (robustness).

#### **4.3.1. Tính dễ học**

Tính dễ học là một đặc trưng quan trọng của HTTT. Nó cho phép người dùng mới thiếu kinh nghiệm có thể sử dụng và sử dụng đạt hiệu quả tối đa. Tính dễ học thể hiện ở một số điểm chính sau:

- Tính dự đoán.
- Tính tổng hợp.
- Tính thân thiện.
- Tính khai quát.
- Tính nhất quán.

### Tính dự đoán

Tính dự đoán của một hệ thống tương tác có nghĩa là dựa vào tri thức đã biết, người dùng có thể xác định được các tương tác mới và đoán nhận kết quả. Tính dự đoán được phân chia thành nhiều cấp độ khác nhau. Tri thức có thể hạn chế chỉ cảm nhận thông tin hiện có và người dùng không cần nhớ những thông tin khác, ngoại trừ thông tin mà họ quan sát được. Việc yêu cầu tri thức sẽ được tăng cường nhằm phát hiện vị trí mà người dùng đang đứng nhằm nhớ lại mọi hành động đã thực hiện và mọi thông tin đã quan sát được trên màn hình chẳng hạn.

Tính dự đoán của một hệ thống tương tác khác với trạng thái xác định của một hệ thống máy tính đơn lẻ. Phần lớn các hệ thống máy tính là các hệ thống với hành vi xác định: nó cung cấp trạng thái ở một thời điểm nào đó và thao tác được thực hiện ở thời điểm này. Tính dự đoán là một khái niệm lấy người dùng làm trung tâm, nó là một hành vi xác định từ sự cảm nhận của người dùng. Thí dụ một trò chơi toán học phổ biến là tìm một dãy số hoàn chỉnh khi biết một vài con số trong dãy. Rõ ràng là dãy số này phải tạo ra bởi một qui luật xác định bởi một hàm hay một thuật toán duy nhất nào đó. Chúng ta biết hàm này và mọi cái chúng ta biết là tính mấy số đầu. Hàm này là xác định và việc kiểm tra dự đoán của chúng ta là phải xem các số tiếp theo.

Một thí dụ khác phức tạp hơn là hình dung rằng bạn vừa tạo ra một bức tranh phức tạp bằng một công cụ đồ họa ngày hôm nay và vài ngày sau bạn quay lại để sửa một chút bức tranh đó. Để chỉnh sửa bạn dùng chuột để chọn đối tượng trên tranh. Bạn có thể tập đối tượng được chọn là gì? Hơn nữa, bạn có thể xác định vùng nào trên màn hình thuộc về đối tượng nào nhất là khi nó chồng lên nhau?

Khái niệm về tính dự đoán thể hiện khả năng của người dùng nhằm xác định tác động của các thao tác với hệ thống. Một dạng khác của tính dự đoán với người dùng là xem thao tác nào có thể thực hiện.

### Tính tổng hợp

Tính dự đoán chỉ tập trung vào khả năng xác định các tác động tiếp theo dựa vào cái đã qua, thông qua mô hình suy nghĩ của con người. Tính dự đoán không chỉ ra cách người dùng hình thành một mô hình về hành vi của hệ thống. Điều quan trọng với người dùng là khảng định hệ quả của một chuỗi các tương tác trước nhằm hình thành nên một mô hình trạng thái của hệ thống.

Khi một thao tác thay đổi một vài sắc thái của một trạng thái trong, điều quan trọng là liệu người dùng có nhận ra không? Nguyên tắc về tính trung thực liên quan tới khả năng giao tiếp của người dùng nhằm cung cấp một số lượng thông tin có tính quan sát được của một thay đổi như vậy. Trong tình huống tốt nhất, sự nhận thức này có thể ngay lập tức,

không đòi hỏi gì thêm. Hoặc trong tình huống xấu nhất, nhận thức này cũng xuất hiện một cách ngẫu nhiên sau khi đã báo cho người dùng bằng cách tạo nên các thay đổi trong quan sát.

Thí dụ về sự khác nhau này có thể dễ nhận ra khi chúng ta chuyển 1 tệp từ thư mục này đến thư mục kia thông qua điều khiển trực tiếp kiểu gấp thả hay sử dụng ngôn ngữ dòng lệnh. Trong giao tiếp gấp thả, chúng ta nhận ra ngay sự thay đổi này, còn trong giao tiếp dòng lệnh, chúng ta phải nhớ thư mục đích và kiểm tra nội dung của nó.

Vấn đề của nguyên tắc trung thực ngẫu nhiên là ở chỗ người dùng phải biết tìm kiếm sự thay đổi. Bạn đọc có thể tìm thấy nhiều ví dụ về ích lợi của nguyên tắc trung thực tức thì so với trung thực ngẫu nhiên. Việc tìm một vài thí dụ coi như một bài tập nhỏ tự làm (một số chức năng của các hệ soạn thảo văn bản).

#### Tính thân thiện

Người dùng thường mang kinh nghiệm của mình trong một lĩnh vực nào đó sang một hệ thống mới. Kinh nghiệm này do tích luỹ hoặc do tương tác với thế giới thực hay với một hệ thống máy tính khác. Với người dùng mới, tính thân thiện của một hệ thống tương tác được đo bởi sự tương hỗ giữa tri thức đang tồn tại với tri thức cần có để thực hiện tương tác có hiệu quả. Thí dụ như trong một hệ soạn thảo văn bản, sự tương tự giữa xử lý bàn phím và máy chữ đã tạo nên một công nghệ mới, có thể thâm nhập ngay lập tức cho những người có chút kinh nghiệm với soạn thảo văn bản, nhưng lại rất ít kinh nghiệm với việc đánh máy. Tính thân thiện gây nên ấn tượng trước tiên của người dùng với hệ thống. Trong trường hợp này, chúng ta quan tâm trước hết tới việc hệ thống được cảm nhận như thế nào, và tiếp theo người dùng có thể xác định khởi tạo một tương tác ra sao?

Một số nhà tâm lý chỉ ra rằng có những tính chất quan trọng của môi đồi tượng trực quan là nó gợi ý cho chúng ta cách thức quản lý. Dáng vẻ của đồi tượng này kích thích tính thân thiện với hành vi của chúng. Thí dụ một phim trên bàn phím gợi ý rằng nó có thể được nhấn. Trong giao tiếp người dùng đồ họa, một phím lệnh gợi ý rằng nó có thể được nhấn bằng chuột (nó không gợi ý sẽ được nhấn như thế nào).

#### Tính khái quát

Người dùng thường có ý mở rộng tri thức của mình từ một ứng dụng cụ thể sang một tình huống tương tự, song chưa gấp bao giờ. Tính khái quát hóa của một hệ thống tương tác hỗ trợ hoạt động này, dẫn đến mô hình dự đoán hoàn thiện hơn cho người dùng.

Tính khái quát có thể xảy ra trong ứng dụng đơn lẻ hoặc xuyên qua một vài ứng dụng. Thí dụ trong một gói đồ họa, một hình vuông được vẽ như một hình chữ nhật có ràng buộc, hay trong một hệ thống window, thao tác “cắt-dán” có thể áp dụng cho mọi đồi tượng buộc, (văn bản, đồ họa, công thức), trong các ứng dụng khác nhau: WORD, PaintBrush và ngược lại.

#### Tính nhất quán

Tính nhất quán có mối quan hệ với tính tương tự trong hành vi xuất hiện trong các tình huống tương tự hay các nhiệm vụ tương tự và là nguyên tắc thường hay được đề cập

đến trong các tài liệu về HCI. Đảm bảo tính nhất quán “be consistent” luôn thúc giục chúng ta. Tuy nhiên, khó khăn của việc thiết kế có tính nhất quán là nó thể hiện ở nhiều dạng khác nhau. Nó không phải là một thuộc tính đơn của một hệ thống tương tác mà nó đáp ứng hay không. Nó phải được áp dụng trong nhiều tình huống như: cách đặt tên, cách cung cấp tham số cho lệnh...

Tính nhất quán cũng cần phải định nghĩa nhằm đảm bảo các nguyên tắc khác mà đôi khi nguyên tắc này lại làm giảm chất lượng của tính nhất quán. Do vậy, tính thân thiện cũng có thể coi là tính nhất quán khi chuyển từ kinh nghiệm của thế giới thực, và tính khái quát cũng coi như tính nhất quán khi tôn trọng kinh nghiệm trên cùng một ứng dụng hay trên tập các môi trường khác nhau.

Tính nhất quán có thể được biểu diễn theo thuật ngữ của dạng biểu diễn đầu vào và đầu ra khi tôn trọng ngữ nghĩa của hành động trong một vài mô hình khái niệm của hệ thống. Thí dụ, trước khi giới thiệu tường minh các phím mũi tên, một vài hệ soạn thảo văn bản sử dụng vị trí tương đối của phím trên bàn phím để chỉ hướng thao tác (sang trái, sang phải, lên hay xuống).

Nhìn chung, tính nhất quán là một nguyên tắc khó dùng, đôi khi còn nguy hiểm. Thí dụ như trước đây khi thiết kế bàn phím máy chữ, người ta cho rằng nên bố trí các phím theo trật tự chữ cái. Tuy nhiên, người ta thấy rằng điều này bất lợi cho việc biểu diễn và gây mệt mỏi cho người dùng. Việc thiết kế bàn phím kiểu QWERTY hay DVORAC là một chủ đề gây nhiều tranh cãi trong thiết kế bàn phím [2].

### 4.3.2. Tính mềm dẻo

Tính mềm dẻo liên quan đến nhiều cách mà người dùng và máy trao đổi thông tin cho nhau. Người ta đưa 5 nguyên tắc để thể hiện tính mềm dẻo:

- Đồi thoại chủ động.
- Đa luồng.
- Di trú nhiệm vụ.
- Thay thế.
- Cá nhân hoá.

### Đồi thoại chủ động

Khi xem xét việc tương tác giữa người dùng và máy tính như đồi thoại giữa các đối tác, điều quan trọng cần biết là đồi tượng nào sẽ chủ động trong giao tiếp. Hệ thống có thể chủ động mọi đồi thoại và trong trường hợp này, người dùng đơn thuần là đáp ứng các yêu cầu đó. Chúng ta gọi kiểu đồi thoại này ưu tiên hệ thống. Thí dụ, một hộp hội thoại kiểu modal (thể thức) ngăn cấm người dùng đưa thông tin trực tiếp vào. Ngược lại, người dùng hoàn toàn có thể chủ động trong giao tiếp và kiểu giao tiếp này gọi là ưu tiên người dùng. Theo cách nhìn của người dùng, đồi thoại hướng hệ thống ngăn cản tính mềm dẻo, còn kiểu đồi thoại hướng người dùng làm tăng tính mềm dẻo.

Một đồi thoại hướng người dùng hoàn toàn cho phép người dùng cung cấp mọi hành động đầu vào ở bất kỳ thời điểm để làm tăng tính mềm dẻo. Tuy nhiên, đó cũng không phải là tình huống hoàn toàn mong đợi, vì nó sẽ làm tăng tính tương tự mà người dùng sẽ mất đi vết các nhiệm vụ mà anh ta chủ động mà còn chưa được hoàn thành.

### **Đa luồng**

Một luồng trong đối thoại là một tập con cố kết của đối thoại đó. Trong đối thoại người dùng-hệ thống, một luồng là một phần của đối thoại có liên quan đến một nhiệm vụ đã cho của người dùng.. Đối thoại đa luồng trong đối thoại người dùng-hệ thống cho phép tương tác hỗ trợ nhiều hơn một nhiệm vụ tại một thời điểm. Đối thoại đa luồng tương tranh cho phép nhiều truyền thông diễn ra cùng một lúc với nhiều nhiệm vụ khác nhau. Đối thoại đa luồng đan xen cho phép việc gối lên nhau tạm thời của các nhiệm vụ với điều khiển là tại một thời điểm chỉ có đối thoại với một nhiệm vụ.

Đa thể thức của một đối thoại liên quan đến đa luồng. J. Coutaz đã đặc trưng hai thứ nguyên của một đối thoại đa thể thức. Thứ nhất, chúng ta xem xét các thể thức riêng biệt được tổ hợp lại ra sao để tạo nên một biểu diễn đầu ra/đầu vào đơn. Đa kênh có thể cùng hiện diện, tuy nhiên một biểu diễn chỉ dành cho một kenh. Thí dụ, để mở một cửa sổ, người dùng có thể lựa chọn việc nhấn kép phím chuột trên một biểu tượng hay dùng một phím nóng hoặc nói “hãy mở cửa sổ”. Ngược lại, một biểu diễn đơn có thể tạo ra bằng việc trộn các kenh. Thí dụ, các cảnh báo của hệ thống thường biểu diễn bởi một cửa sổ cảnh báo kèm lời cảnh báo hay tiếng bip bip của hệ thống.

Một hệ thống cửa sổ hỗ trợ một cách tự nhiên một đối thoại đa luồng đan xen giữa các nhiệm vụ phủ lên nhau. Mỗi một cửa sổ biểu diễn một nhiệm vụ khác nhau: cái thì biểu diễn văn bản, cái thì cho phép tìm tệp, cái khác thì cho phép nghe nhạc, . . .

### **Di trú nhiệm vụ**

Di trú nhiệm vụ liên quan đến việc chuyển điều khiển thực hiện nhiệm vụ giữa người dùng và hệ thống. Kiểm tra lỗi chính tả của một văn bản là một thí dụ tốt về tính cần thiết của nguyên tắc di trú nhiệm vụ.

Tuy nhiên, trong các ứng dụng có tính an toàn cao như trong điều khiển bay, tính di trú có thể làm giảm tính có về thật của tai nạn. Trên bảng điều khiển của buồng lái, rất nhiều điều khiển có thể thực hiện mà một người lái có thể bị ngợp nếu phải xử lý tất cả. Nhìn chung, máy bay có thể được điều khiển hoàn toàn tự động, song trong trường hợp khẩn cấp, người lái có thể can thiệp tức thì, tức là giành quyền điều khiển.

### **Thay thế**

Một tập giá trị tương đương có thể thay thế cho nhau tùy ứng dụng. Khi xác định là một văn bản, người dùng có thể chọn đơn vị là inch, centimet, song cũng có thể là điểm. Hơn nữa việc cung cấp giá trị có thể là tường minh hay thông qua tính toán. Đa biểu diễn cũng được xem như là một sự thay thế. Thí dụ, ta có thể biểu diễn nhiệt độ bằng nhiệt kế số hay đồ thị, tùy theo mức độ quan trọng nào.

Cơ hội tương đương cũng là một dạng thể hiện của tính thay thế. Người dùng có thể chọn biểu diễn vào khác biểu diễn ra hay sử dụng ra để biểu diễn vào.

### **Cá nhân hoá**

Cá nhân hoá là một cách thức giao tiếp người dùng bởi người dùng và hệ thống. Về khía cạnh hệ thống, chúng ta không quan tâm các thay đổi có chủ ý từ người lập trình làm

## **Chương 4: Thiết kế giao tiếp người - máy**

thay đổi hệ thống và giao diện trong quá trình duy trì hệ thống. Chúng ta chỉ quan tâm đến các thay đổi tự động mà hệ thống muốn thực hiện được dựa vào tri thức của nó về người dùng. Chúng ta phân biệt cái thay đổi của hệ thống với nhiều người dùng với ưu tiên hệ thống: tính thích nghi và tính có thể thích nghi.

Tính thích nghi liên quan đến khả năng của người dùng điều chỉnh dạng đầu vào/đầu ra. Tính cá nhân hóa có thể rất hạn chế. Thí dụ, người dùng chỉ có thể thay đổi tên, vị trí phím lệnh, kiểu này gọi là các nhân hóa từ vựng, bị hạn chế bởi bề mặt giao diện và không làm thay đổi cấu trúc chung của giao diện. Tính có thể thích nghi là cá nhân hóa tự động về giao diện người dùng bởi hệ thống.

### **4.3.3. Tính vững chắc**

Người dùng dùng máy tính nhằm thực hiện một số các mục đích trong công việc hoặc trong lĩnh vực nhiệm vụ. Tính vững chắc của một tương tác bao trùm các đặc trưng hỗ trợ việc thực hiện thành công và khẳng định đích. Tính vững chắc thể hiện ở một số tiêu chí sau:

- Tính quan sát (Observability).
- Tính khôi phục (Recoverability).
- Tính đáp ứng (Responsiveness).
- Tính tương hợp nhiệm vụ (Task conformance).

#### **Tính quan sát**

Tính quan sát được cho phép người dùng đánh giá trạng thái bên trong của hệ thống nhờ biểu diễn cảm nhận được trên giao tiếp. Tính quan sát được thể hiện qua 5 tiêu chí sau:

- Quét tuần tự.
- Ngầm định.
- Đạt được.
- Ôn định.
- Nhìn thấy thao tác.

#### **Tính khôi phục**

Người dùng có thể gây lỗi trong giao tiếp và họ 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. Tính khôi phục có thể đi theo hai hướng: tiến lên hay quay lui. Khôi phục kiểu tiến lên là sự chấp nhận trạng thái hiện tại và đàm phán từ trạng thái đó tiến về trạng thái đích. Khôi phục tiến là khả năng duy nhất nếu ảnh hưởng của tương tác không thể huỷ bỏ được. Khôi phục quay lui là thử bỏ đi ảnh hưởng của các tương tác trước nhằm quay lại trạng thái trước khi xử lý.

Trong một hệ soạn thảo văn bản, do ấn nhầm một phím, một đoạn văn bản bị xoá đi, chương trình có thể khôi phục bằng việc nhấn phím “undo”.

Việc khôi phục có thể khởi xướng bởi người dùng hoặc hệ thống. Khi thực hiện bởi hệ thống, việc khôi phục, việc khôi phục liên quan đến khái niệm dung lỗi mà ta thường gặp trong CNPM. Tuy nhiên, trong CNPM việc dung lỗi chỉ liên quan đến chức năng của hệ

## **Chương 4: Thiết kế giao tiếp người - máy**

thống mà không gắn với chủ ý người dùng. Khi khôi phục lỗi theo đề xuất của người dùng, điều quan trọng là phải xác định chủ ý của hành động khôi phục của người dùng, có nghĩa là họ muốn tiến hay lui.

Tóm lại, để cung cấp khả năng khôi phục thì khôi phục phải phản ánh công việc được làm hay không được làm. Nguyên tắc hệ quả tương ứng phát biểu rằng nếu khó thực hiện “undo” thì trước hết cũng phải khó thực hiện trạng thái đó.

### **Tính đáp ứng**

Tính đáp ứng đo vận tốc giao tiếp giữa người dùng và hệ thống. Thời gian đáp ứng, nói chung được định nghĩa như là khoảng thời gian cần thiết để biểu diễn thay đổi trạng thái. Thường người ta mong muốn thời gian đáp ứng là ngắn nếu không nói tức thì. Tuy nhiên, nếu đáp ứng không tức thì, hệ thống cần có một vài chỉ thị để người dùng hiểu là hệ thống đã nhận được yêu cầu và đang xử lý nó.

### **Tính tương hợp nhiệm vụ**

Khó có thể nói là hệ thống máy tính cài đặt đủ tập các dịch vụ tính toán như nó được chỉ ra ở giai đoạn đặc tả. Điều chủ yếu là hệ thống cho phép người dùng thực hiện bất cứ nhiệm vụ mong muốn nào trong một lĩnh vực công việc cụ thể như đã xác định trong bước phân tích nhiệm vụ. Tính đầy đủ của nhiệm vụ liên quan đến mức độ mà các dịch vụ của hệ thống có thể ảnh xạ vào mọi nhiệm vụ của người dùng. Tuy nhiên, rất có thể cách nhìn của các công cụ dựa máy tính sẽ gợi ý cho người dùng một số nhiệm vụ còn chưa cảm nhận được. Vì thế, người ta mong muốn các dịch vụ hệ thống là đa năng sao cho người dùng có thể định nghĩa các nhiệm vụ mới.

Việc tranh luận về tính tương hợp nhiệm vụ xuất phát từ việc hiểu thành công của giao tiếp điều khiển trực tiếp. Chúng ta có thể coi giao tiếp điều khiển trực tiếp như là một thế giới tách riêng với cái bên trong của hệ thống. Tính đầy đủ của hệ thống chỉ là một phần của tính tương hợp.

### **4.4. Thiết kế giao tiếp người dùng - máy tính**

CNPM cung cấp phương tiện để hiểu cấu trúc của quá trình thiết kế và quá trình này có thể khẳng định tính hiệu quả trong thiết kế HCI. CNPM đã bao trùm lên cả vấn đề tương tác. Rất nhiều sách đã đề cập đến quy trình thiết kế chung hay vòng đời phần mềm. Để dễ theo dõi, trong phần này, chúng ta chỉ khái quát lại một số điểm thể hiện vòng đời phần mềm và tập trung chính vào đặc thù riêng của quá trình thiết kế giao tiếp người dùng - máy tính. Chúng ta sẽ đề cập tới một số nội dung sau:

- 1) Tổng quan về quá trình thiết kế
- 2) Vòng đời và vòng đời hình sao trong thiết kế giao tiếp người dùng - máy tính
- 3) Thiết kế lặp và mẫu thử

#### **4.4.1. Tổng quan về quá trình thiết kế**

Trong phần đầu chương, ta đã tập trung xem xét hai sắc thái quan trọng của tính tiện dụng của các hệ thống tương tác bằng việc xem xét các mô thức của tương tác thành công, các nguyên tắc và các chuẩn cho quá trình thiết kế.

## Chương 4: Thiết kế giao tiếp người - máy

Ngay cả khái niệm thiết kế cũng có nhiều cách hiểu khác nhau [1 trang 352], song về cơ bản chúng ta hiểu rằng: thiết kế liên quan đến quá trình phát triển một sản phẩm, một hệ thống và các cách biểu diễn khác nhau của một hệ thống sẽ được tạo ra trong quá trình đó.

Nhiều khái niệm và quy trình thiết kế được chỉ ra trong các sách về CNPM và nó cũng được dùng trong thiết kế các hệ thống tương tác như vòng đời (life cycle), mô hình,... Tuy nhiên, mục đích của thiết kế các hệ thống tương tác là phải đảm bảo tính tiện dụng tối đa, do vậy một số qui trình trong CNPM dùng trong thiết kế có thể thay đổi (xem phần 4.5.2). Một trong những thay đổi là thiết kế có tính lặp: thiết kế không chỉ diễn ra trong một giai đoạn mà trong suốt vòng đời của quá trình phát triển.

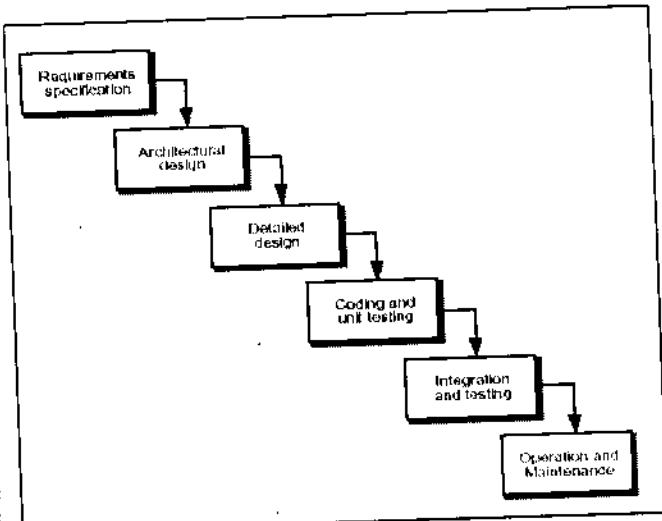
Đầu ra của thiết kế là các tài liệu mô tả dưới dạng văn bản hay đồ họa (dưới dạng các lưu đồ/sơ đồ) đặc tả yêu cầu người dùng, phân tích nhiệm vụ, mô tả đối thoại,...

### 4.4.2 Vòng đời và vòng đời hình sao trong thiết kế giao tiếp người-máy

Trong CNPM, quan điểm chung nhất cho rằng quá trình phát triển hệ thống phần mềm bao gồm nhiều giai đoạn: đặc tả yêu cầu phần mềm, thiết kế hệ thống, lập trình, kiểm thử và cuối cùng là bảo trì khai thác phần mềm. Khái niệm vòng đời phần mềm là quãng thời gian bắt đầu có yêu cầu xây dựng phần mềm đến khi có phần mềm, phần mềm được khai thác rồi chết đi. Quan niệm này được thể hiện dựa trên mô hình thác nước (water model) như chỉ ra trên hình 4.1.

Mô hình thác nước cho rằng việc phát triển hệ thống phần mềm được tiến hành qua nhiều giai đoạn, song cơ bản là tuyến tính. Điều này có nghĩa các giai đoạn được thực hiện theo một trình tự hết cái này đến cái khác.

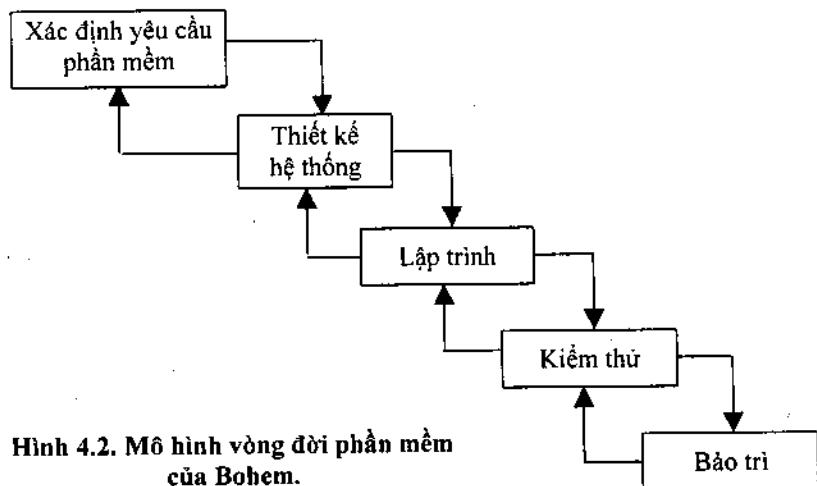
Tuy nhiên, thực tế chỉ ra rằng điều này là không chính xác. Thứ nhất, ranh giới giữa các giai đoạn không rõ ràng. Thứ hai, thường có phản hồi từ các giai đoạn tiếp sau về giai đoạn trước (hình 4.2).



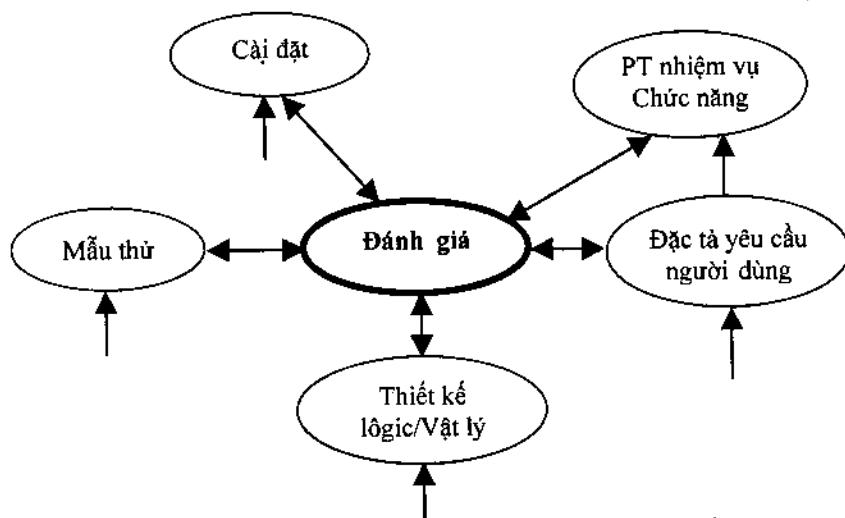
Hình 4.1. Mô hình thác nước của vòng đời phần mềm.

Nguyên tắc thiết kế lấy người dùng làm trung tâm đã coi người dùng là mục đích của thiết kế, là một thành viên trong quá trình thiết kế. Chính nguyên tắc này đã dẫn đến khái niệm vòng đời hình sao trong thiết kế giao tiếp người dùng-máy tính do Hix & Hartson đề xuất năm 1993.

## Chương 4: Thiết kế giao tiếp người - máy



Hình 4.2. Mô hình vòng đời phần mềm của Boehm.



Hình 4.3. Vòng đời hình sao của HCI (Hix & Hartson, 1993).

Người dùng không chỉ đơn thuần là người bình luận về ý tưởng của người thiết kế mà họ còn tham gia vào mọi khía cạnh của quá trình thiết kế. Thiết kế thực sự phải là thiết kế lặp.

Thiết kế phải tích hợp được tri thức của người dùng con người, các chuyên gia từ nhiều lĩnh vực. Việc kiểm thử, đánh giá phải được thực hiện ngay trong quá trình thiết kế. Trong quá trình thiết kế, người ta sử dụng rất nhiều mô hình khác nhau. Mô hình không chỉ hỗ trợ quá trình thiết kế mà còn cho cả quá trình kiểm thử và đánh giá.

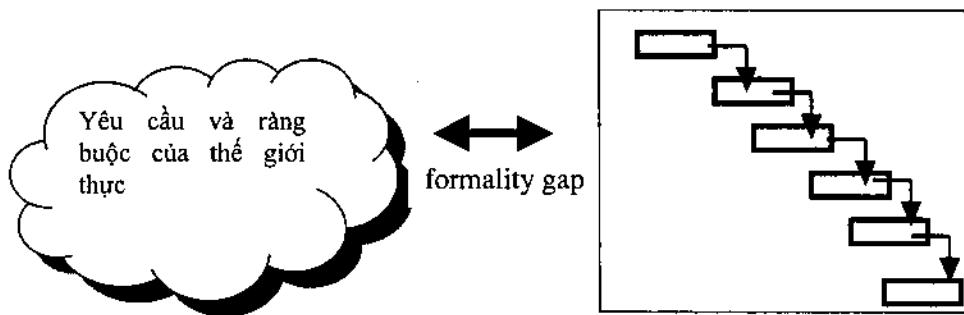
## **Chương 4: Thiết kế giao tiếp người - máy**

### **4.4.3. Hợp thức và kiểm chứng**

Xuyên suốt vòng đời phát triển, người thiết kế phải kiểm tra để đảm bảo rằng việc thiết kế đáp ứng yêu cầu đã thỏa thuận với người dùng. Việc kiểm tra này liên quan đến tính hợp thức (validation) và kiểm chứng (testing). Theo Boehm, tính hợp thức bảo đảm làm đúng sản phẩm, còn kiểm chứng đảm bảo sản phẩm tốt.

Nhiều ngôn ngữ được dùng trong suốt quá trình thiết kế: từ ngôn ngữ tự nhiên hình thức đến các ngôn ngữ toán học chính xác. Tính hợp thức và kiểm chứng rất khó thực hiện trong ngôn ngữ và không nói là không thể khi thử giữa các ngôn ngữ.

Kiểm chứng chỉ diễn ra trong một hoạt động đơn của vòng đời hay giữa hai hoạt động liên tiếp. Tính hợp thức của thiết kế chỉ ra rằng bên trong các hoạt động khác nhau, yêu cầu của người dùng được đáp ứng. Hợp thức là nhiệm vụ chủ quan hơn kiểm chứng. Một câu hỏi cho cả hợp thức và kiểm chứng là chính xác điều gì sẽ cấu thành nên một bằng chứng? Chúng ta có thể nói đó là ngôn ngữ dùng trong thiết kế và những điều cơ bản về ngữ nghĩa của ngôn ngữ đó.



**Hình 4.4. Mô phỏng kiểu hình thức gián đoạn.**

Trong quá trình mô hình hóa thế giới thực với các giai đoạn của thiết kế dễ thể hiện tính không tương ứng. Vì vậy người ta hay dùng kiểu hình thức gián đoạn (formality gap) như chỉ ra trên hình 4.4. Hình thức gián đoạn có nghĩa là tính hợp thức luôn liên quan đến một vài mở rộng ngữ nghĩa chủ quan của bằng chứng.

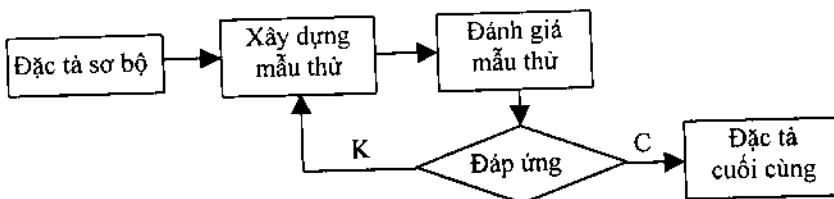
### **4.5. Thiết kế lắp và mẫu thử**

Các đặc tả yêu cầu người dùng hiếm khi đầy đủ do nguyên nhân này hay nguyên nhân khác. Hơn nữa quá trình đặc tả yêu cầu lại diễn ra ở giai đoạn đầu của vòng đời nên thường phải được hiệu chỉnh trong quá trình thiết kế. Do vậy, cách tốt nhất để đảm bảo các đặc trưng của thiết kế là phải xây dựng, kiểm thử và đánh giá với người dùng thực sự. Sau đó, thiết kế cần phải được hiệu chỉnh để sửa các lỗi phát hiện trong kiểm thử. Đó chính là quá trình thiết kế lắp, một quá trình thiết kế nhằm khắc phục tính cố hữu của đặc tả không đầy đủ.

## Chương 4: Thiết kế giao tiếp người - máy

Về khía cạnh kỹ thuật, quá trình thiết kế lặp được mô tả bằng việc sử dụng các mẫu thử (prototype). Các mẫu thử là sự bắt chước hay mô phỏng một số chức năng đặc trưng chủ không phải của một hệ thống đầy đủ (hệ thống có thể chưa tồn tại). Có ba kỹ thuật mẫu thử :

- Tung gia (Throw away): Mẫu thử được xây dựng và kiểm thử. Tri thức thiết kế thu được từ cuộc tập duyệt này sẽ có ích cho việc xây dựng hệ thống cuối cùng, nhưng mẫu thử hiện thời sẽ bị huỷ (hình 4.5a).
- Gia tăng (Incremental): sản phẩm cuối cùng được xây dựng như các thành phần riêng biệt, mỗi thành phần ở một thời điểm. Đó là một thiết kế toàn bộ cho hệ thống cuối cùng, nhưng được phân thành các thành phần nhỏ và độc lập. Sản phẩm cuối cùng được cung cấp như một chuỗi sản phẩm, một chuỗi con cung cấp bao gồm nhiều hơn một thành phần (hình 4.5b)
- Tiến hoá (Evolutionary): Trong cách tiếp cận này, mẫu thử không bị huỷ bỏ và được dùng như cơ sở cho lần lặp tiếp theo. Như vậy, hệ thống hiện thời được xem như là sự tiến hoá từ phiên bản rất thô ban đầu để sản phẩm cuối cùng (hình 4.5c).



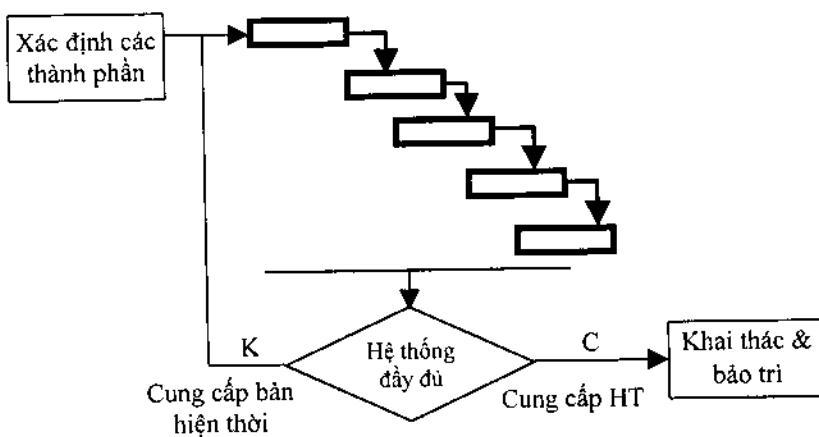
Hình 4.5a. Mẫu thử kiểu Throw away.

Nhìn chung, mẫu thử khác sản phẩm cuối cùng ở một số chức năng cũng như tính hiệu quả. Do vậy, việc sử dụng mẫu thử thể hiện một số ưu việt:

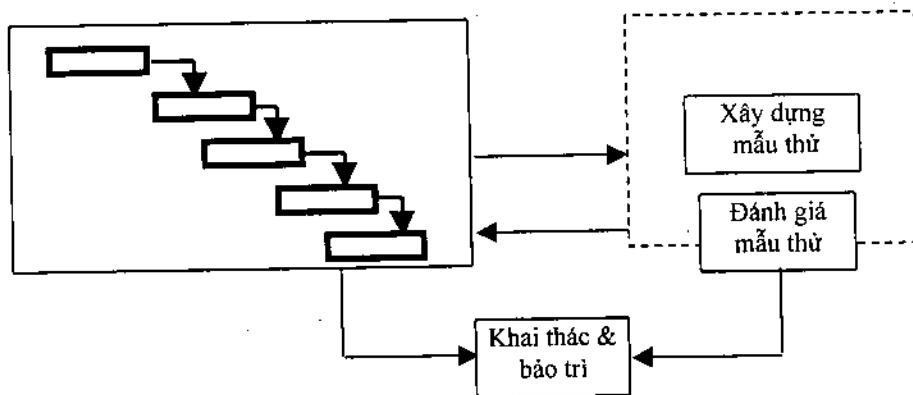
- Làm mịn đặc tả.
- Làm mịn thiết kế.
- So sánh đánh giá các thiết kế.
- Chứng minh cho một ý tưởng.
- Cho các đối tượng: người dùng, nhà thiết kế, ...

Tuy nhiên, mẫu thử cũng bộc lộ một số hạn chế:

- Tốn thời gian: Xây dựng mẫu thử cũng cần có thời gian. Nếu là mẫu thử kiểu tung gia, mỗi lần xong ta lại huỷ đi nên lãng phí thời gian.
- Kế hoạch: Phần lớn các nhà quản lý dự án không có kinh nghiệm cần thiết để lập kế hoạch phù hợp và chi phí cho một quá trình thiết kế.
- Đặc trưng phi chức năng: Thường cái quan trọng nhất của hệ thống là các đặc trưng phi chức năng như: tính an toàn, độ tin cậy..., cần phải được đảm bảo trong quá trình thiết kế.



**Hình 4.5b. Mẫu thử kiểu Incremental.**



**Hình 4.5c. Mẫu thử kiểu Evolutionary.**

### Một số kỹ thuật xây dựng mẫu thử

#### 1) Storyboard

Là kỹ thuật minh họa các hình ảnh chụp nhanh của hệ thống, có nguồn gốc từ hoạt hình và công nghệ làm phim. Kỹ thuật này dùng để minh họa một kịch bản với các lựa chọn cố định hoặc khá hạn chế. Kỹ thuật này thể hiện một số ưu điểm chính sau:

- Không nhất thiết dùng máy tính.
- Minh họa nhanh ý tưởng, do vậy có ích trong giai đoạn thiết kế.
- Hình dung các ý tưởng mới không hỗ trợ bởi các công cụ đã có hay công nghệ còn chưa tồn tại. Tuy nhiên cần có vai trò giám sát của người thiết kế.

## 2) Mô phỏng chức năng hạn chế

Nhiều chức năng cần được xây dựng trong mẫu thử để minh họa công việc mà ứng dụng cần thực hiện. Tuy nhiên, storyboard cũng như hoạt hình không đủ khả năng. Do vậy một số chức năng cần phải được mô phỏng bởi nhóm thiết kế. Công cụ lập trình có thể hỗ trợ. Người thiết kế cần xây dựng các đối tượng đồ họa hay văn bản và gắn vào đó để kiểm tra một số chức năng của hệ thống. Một minh họa đơn giản sẽ được chỉ ra trong phần thí dụ.

### Một số công cụ

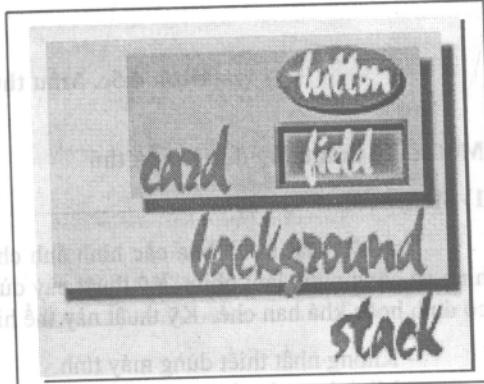
Nhiều công cụ hỗ trợ cho việc xây dựng mẫu thử đã được phát triển. Trong số đó phải kể đến các ngôn ngữ lập trình như: Visual Basic, Thành phần Swing của Java và một số công cụ khá nổi tiếng khác như Hypercard, Revolution. Dưới đây, chúng ta chỉ đề cập tóm tắt hai công cụ, còn các ngôn ngữ lập trình bạn đọc tự tìm hiểu. Phần chi tiết tìm trong tài liệu tham khảo/phụ lục.

#### 1) Hypercard

Hypercard là một gói công cụ phần mềm được viết để tạo các ứng dụng tương tác đa phương tiện cho dòng máy Macintosh do Bill Atkinson, Công ty Apple đề xuất năm 1987. Sử dụng Hypercard có thể tạo ra nhiều ứng dụng khác nhau từ xử lý văn bản, cơ sở dữ liệu đến các bảng tính. Hai thành phần chính của Hypercard là Stack và card, giống như quyển sách điện tử và các trang. Một Stack bao gồm nhiều card và các card có thể sắp xếp theo trình tự khác nhau tùy theo ý đồ của người dùng. Các card có thể bổ sung vào Stack hay loại ra khỏi Stack.

Mỗi card trong Hypercard đều có hai lớp. Lớp trên cùng gọi là card level, lớp dưới gọi là lớp nền. Một điều đặc biệt của lớp nền là mọi card đều dùng chung mọi đặc điểm trên nền. Mọi thứ trên lớp card level chỉ được nhìn thấy trên lớp card của card mà nó xuất hiện trên đó. Mỗi card gồm: các đối tượng đồ họa, các trường (field) cho văn bản và các phím lệnh để liên kết các card. Kiến trúc các lớp của hypercard được chỉ ra trên hình 4.6.

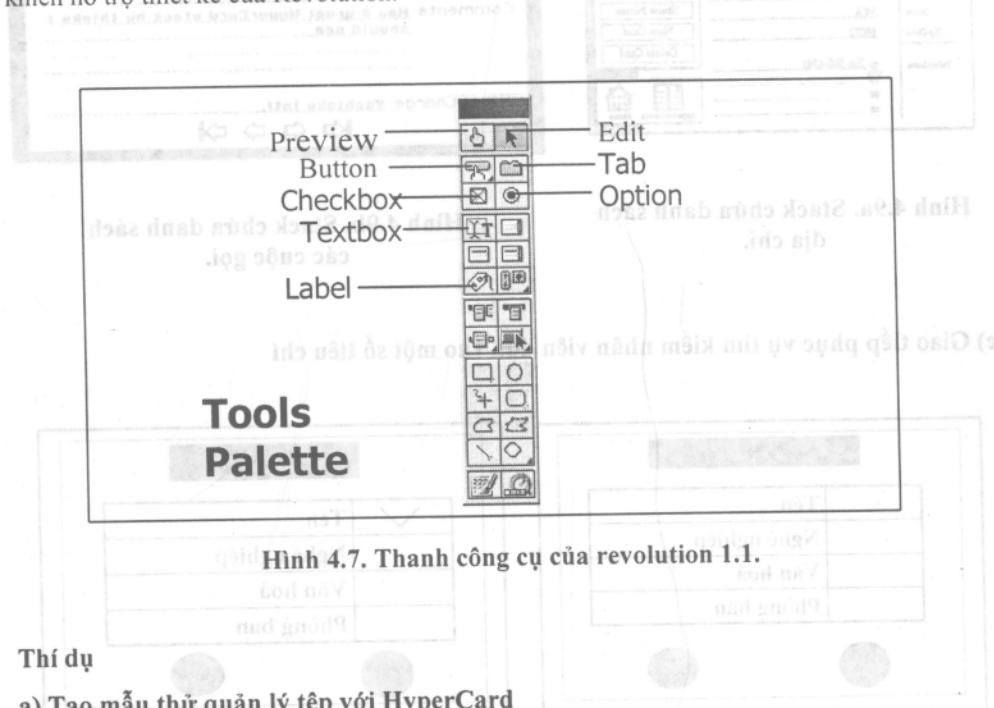
Các tính năng của Hypercard bao gồm: Hypertext, cơ sở dữ liệu và các công cụ thiết kế giao diện. Hypercard sử dụng ngôn ngữ HyperTalk, một ngôn ngữ lập trình giống như ngôn ngữ tiếng Anh tự nhiên hướng đối tượng. Ngoài ra, Hypercard hỗ trợ đa phương tiện. Hypercard cho phép liên kết các stack với các câu lệnh đơn để điều khiển việc thực hiện chương trình.



Hình 4.6. Kiến trúc hypercard.

## 2) Revolution

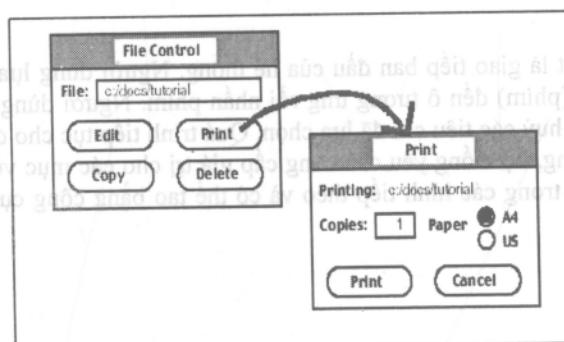
Revolution là một phần mềm thiết kế nhanh mẫu thử giao diện do Runtime Revolution Team phát triển cho các dòng máy PC-IBM. Giống như Hypercard, revolution cung cấp các palette phục vụ cho mô phỏng các điều khiển thiết kế card. Mỗi card có thể gồm các hộp văn bản, các nhãn, các hộp đẩy kéo, hình ảnh,... và nằm trên một stack. Các stack tương tác với nhau qua HyperTalk giống như Hypercard. Hình 4.7 minh họa các điều khiển hỗ trợ thiết kế của Revolution.



**Hình 4.7. Thanh công cụ của revolution 1.1.**

### Thí dụ

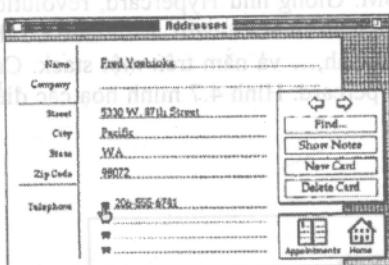
#### a) Tạo mẫu thử quản lý tệp với HyperCard



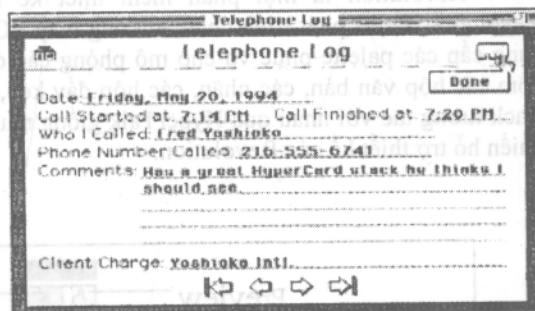
**Hình 4.8. Thiết kế mẫu thử cho in tệp.**

## Chương 4: Thiết kế giao tiếp người - máy

### b) Theo dõi các cuộc gọi điện thoại



Hình 4.9a. Stack chứa danh sách địa chỉ.



Hình 4.9b. Stack chứa danh sách các cuộc gọi.

### c) Giao tiếp phục vụ tìm kiếm nhân viên dựa vào một số tiêu chí

Chọn tiêu chí tìm	
<input type="checkbox"/>	Tên
<input type="checkbox"/>	Nghề nghiệp
<input type="checkbox"/>	Văn hoá
<input type="checkbox"/>	Phòng ban

Chọn tiêu chí tìm	
<input checked="" type="checkbox"/>	Tên
<input type="checkbox"/>	Nghề nghiệp
<input type="checkbox"/>	Văn hoá
<input type="checkbox"/>	Phòng ban

Hình thứ nhất là giao tiếp ban đầu của hệ thống. Người dùng lựa chọn một tiêu chí bằng cách rê chuột (phím) đến ô tương ứng rồi nhấn phím. Người dùng có thể chọn nhiều tiêu chí hay quay lại huỷ các tiêu chí đã lựa chọn. Quá trình tiếp tục cho đến khi người dùng chấp nhận. Cuối cùng, hệ thống yêu cầu cung cấp giá trị cho các mục vừa chọn. Các minh họa này được chi ra trong các hình tiếp theo và có thể tạo bằng công cụ hay ngôn ngữ lập trình.

## Chương 4: Thiết kế giao tiếp người - máy

Chọn tiêu chí tìm	
<input checked="" type="checkbox"/>	Tên
<input type="checkbox"/>	Nghề nghiệp
<input checked="" type="checkbox"/>	Văn hoá
<input type="checkbox"/>	Phòng ban
<b>OK</b>	<b>Back</b>

Chọn tiêu chí tìm	
<input checked="" type="checkbox"/>	Tên
<input type="checkbox"/>	Nghề nghiệp
<input checked="" type="checkbox"/>	Văn hoá
<input type="checkbox"/>	Phòng ban
<b>OK</b>	<b>Back</b>

Chọn tiêu chí tìm	
<input type="checkbox"/>	Tên
<input type="checkbox"/>	Nghề nghiệp
<input checked="" type="checkbox"/>	Văn hoá
<input type="checkbox"/>	Phòng ban
<b>OK</b>	<b>Back</b>

Chọn tiêu chí tìm	
<input checked="" type="checkbox"/>	Tên
<input checked="" type="checkbox"/>	Nghề
<input checked="" type="checkbox"/>	Văn
<input checked="" type="checkbox"/>	Phòngban
<b>OK</b>	<b>Back</b>

Chọn giá trị	
Nghề nghiệp	
Văn hoá	
<b>OK</b>	<b>Back</b>

Trong màn hình nhập liệu, hệ thống sẽ cung cấp các mục được chọn và con trỏ sẽ định vị tại ô trên cùng. Hệ thống cũng cung cấp hai phím lệnh để người dùng có thể yêu cầu thực hiện hay quay lui.

Tóm lại, trong quá trình thiết kế giao tiếp người dùng, mẫu thử là một công cụ quan trọng không chỉ cho nhà thiết kế mà còn cho đánh giá và người dùng. Mẫu thử có thể xây dựng thông qua các công cụ lập trình như Java Swing, VB hay qua một số công cụ khá nổi tiếng như Hypercard, Revolution, FlashGet,...

# 5

## CÁC MÔ HÌNH NHẬN THỨC NGƯỜI DÙNG

### Nội dung

- 5.1. Tổng quan
- 5.2. Mô hình đặc tả yêu cầu người dùng
  - 5.2.1. Mô hình kỹ thuật xã hội (OSTA)/ mô hình thiết kế cộng tác
  - 5.2.2. Mô hình hệ thống phần mềm (Eason, 1992)
  - 5.2.3. Mô hình đa cách nhìn (Multiview)
- 5.3. Mô hình nhận thức
  - 5.3.1. Mô hình phân cấp nhiệm vụ (GOMS)
  - 5.3.2. Mô hình ngôn ngữ
  - 5.3.3. Mô hình Keystroke

## **Chương 5: Các mô hình nhận thức người dùng**

---

### **5.1. Tổng quan**

Trong mọi lĩnh vực kỹ thuật, người thiết kế luôn lựa chọn và sử dụng các mô hình thích hợp trong quá trình thiết kế. Thí dụ như trong CNPM, trong quá trình thiết kế, người ta dùng rất nhiều mô hình như: mô hình thuần thực khả năng CMM, mô hình tuyển tính, mô hình phát triển nhanh RAD,..., tùy theo ứng dụng mà chọn lựa. Một số mô hình mang tính đánh giá, có nghĩa là nó cho ta biết một thiết kế có tính xác đáng hay không, một số mô hình lại mang tính khởi sinh, có nghĩa là nó đóng góp cho chính quá trình thiết kế.

Trong giao tiếp người dùng-máy tính, người ta phân làm hai nhóm mô hình. Nhóm thứ nhất dùng để thiết lập nhu cầu người dùng, gọi là các mô hình đặc tả yêu cầu người dùng. Các mô hình trong nhóm này gồm có:

- Mô hình kỹ thuật xã hội OSTA.
- Mô hình hệ thống phần mềm.

Nhóm thứ hai là nhóm mô hình nhận thức. Nó biểu diễn người dùng trong một hệ thống tương tác, mà mục đích chính là hiểu các sắc thái người dùng trong nhận thức, trong tri thức và trong xử lý. Nhóm này bao gồm các mô hình:

- Mô hình phân cấp: biểu diễn một nhiệm vụ của người dùng và cấu trúc mục đích.
- Mô hình ngôn ngữ : biểu diễn văn phạm người dùng - hệ thống.
- Mô hình GOMS.
- Mô hình KEYTROCKE.

Trong các phần dưới đây, chúng ta lần lượt giới thiệu chi tiết các mô hình, cách áp dụng các mô hình.

### **5.2. Mô hình đặc tả yêu cầu người dùng**

Thâu tóm yêu cầu người dùng là một phần quan trọng trong vòng đời phần mềm, tuy nhiên hành động này chỉ tập trung vào các yêu cầu chức năng - cái mà hệ thống phải thực hiện, hiếm khi chú ý đến các yêu cầu phi chức năng như tính tiện dụng và tính chấp nhận được. Nay cả trong các vấn đề quan tâm, người ta cũng chỉ phản ánh yêu cầu người dùng về khía cạnh quản lý hơn là thu thập thông tin về người dùng. Mô hình người dùng sẽ chú ý làm cân bằng vấn đề này bằng cách coi người dùng là một thành viên của quá trình thiết kế. Họ có mặt ở đó không chỉ để bình luận hay xem xét ý tưởng mà là một thành viên thực sự. Nhiều mô hình và phương pháp được áp dụng để thâu tóm một cách rộng hơn về yêu cầu hệ thống. Trong phần dưới đây chúng ta sẽ đề cập đến ba mô hình:

- Mô hình kỹ thuật xã hội - OSTA.
- Mô hình hệ thống phần mềm - SSM.
- Mô hình đa cách nhìn.

#### **5.2.1. Mô hình kỹ thuật xã hội (OSTA)/ mô hình thiết kế cộng tác**

Mô hình này xem xét sự liên quan đến con người và các vấn đề tổ chức, xã hội, kỹ thuật trong thiết kế. Mô hình chỉ ra rằng công nghệ không bao giờ phát triển một cách biệt lập mà nó gắn với một môi trường xã hội rộng rãi. Dưới đây, ta sẽ xem xét hai mô hình con của mô hình thiết kế cộng tác này là mô hình USTA và OSTA.

### **Mô hình USTA (User skills and Task Analysis)**

USTA là mô hình cho phép đội phát triển hiểu và lập tài liệu đặc tả yêu cầu người dùng. Nó thường sử dụng các lưu đồ để phân tích cùng với các đặc tả bằng ngôn ngữ tự nhiên. Mô hình này thường áp dụng trong các tổ chức nhỏ và tập trung vào đặc tả yêu cầu các cổ đông (stakeholder). Yêu cầu của mọi cổ đông đều được xem xét không chỉ yêu cầu của người dùng cuối.. Có bốn loại cổ đông:

- Loại một: những người dùng hệ thống.
- Loại hai: những người không sử dụng trực tiếp hệ thống song có nhận thông tin từ đầu ra hệ thống.
- Loại ba: không thuộc hai loại trên song có chịu tác động từ sự thành công hay thất bại của hệ thống.
- Loại bốn: là những người tham gia vào quá trình thiết kế, phát triển và bảo trì hệ thống.

Mô hình này thường được áp dụng trong giai đoạn đầu của hệ thống khi mà cơ hội sản xuất được xác định. Nó là phương pháp dựa vào khuôn điền, nhằm cung cấp một tập các câu hỏi áp dụng cho mỗi giai đoạn.

### **Mô hình OSTA (Open System Task Analysis)**

OSTA thử mô tả cái sẽ xảy ra khi một hệ thống kỹ thuật sẽ được đưa vào môi trường kỹ thuật của một tổ chức. Mô hình này do Eason và Harker đề xuất 1998-1999, nhằm hiểu cả hệ thống xã hội cũng như hệ thống kỹ thuật. OSTA theo đuổi một mô hình phân tích hệ thống xã hội, trong đó các yêu cầu kỹ thuật (cấu trúc và chức năng) của hệ thống được đặc tả bên cạnh các yêu cầu xã hội (tính tiện dụng và tính chấp nhận được). OSTA có tám giai đoạn chính:

- i) Nhiệm vụ chính mà công nghệ phải hỗ trợ là xác định mục đích của người dùng.
- ii) Nhiệm vụ đầu vào của hệ thống được xác định gồm nhiều nguồn, nhiều dạng khác nhau.
- iii) Môi trường mà hệ thống sẽ đưa vào cần được xác định về các khía cạnh: vật lý, kinh tế và chính trị.
- iv) Mô tả quá trình phát triển bao gồm mô tả chức năng và thường được biểu diễn dưới dạng lưu đồ.
- v) Phân tích hệ thống xã hội. Vai trò của con người trong tổ chức được đánh giá trong mối quan hệ với những cái khác.
- vi) Hệ thống kỹ thuật được phân tích theo ngữ cảnh hệ thống mới sẽ được tích hợp ra sao với các hệ thống khác cũng như với hệ thống cũ.
- vii) Đáp ứng về hiệu quả so với các yêu cầu đã nhấn mạnh trong một hệ thống xã hội mới.

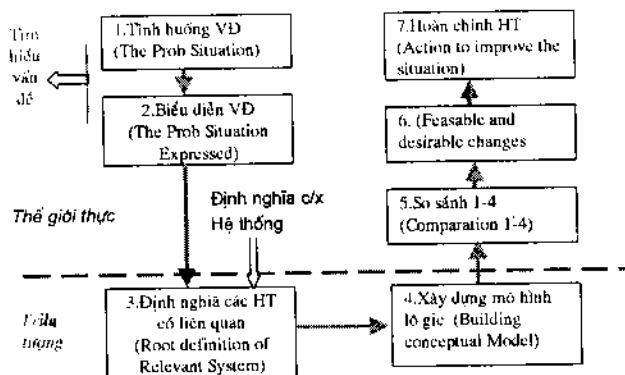
## Chương 5: Các mô hình nhận thức người dùng

viii) Các yêu cầu của hệ thống kỹ thuật mới cần được đặc tả trong giai đoạn phân tích.

Điểm mạnh của mô hình OSTA là sử dụng các ký pháp quen thuộc với nhà thiết kế. Tuy nhiên, nó cũng thể hiện một số hạn chế: cần có chuyên gia để hướng dẫn, mức độ tích hợp với các hệ thống khác và tính tương hợp với tổ chức và xã hội.

### 5.2.2. Mô hình hệ thống phần mềm (Eason, 1992)

Nếu trong mô hình hệ thống xã hội kỹ thuật chúng ta tập trung vào việc xác định yêu cầu của con người và hệ thống thì trong mô hình hệ thống phần mềm - SSM, chúng ta sẽ xem xét theo cách rộng hơn: tổ chức như là một hệ thống mà kỹ thuật và con người là các thành phần. Mô hình hệ thống phần mềm do Checkland và Scholes đề xuất năm 1991 và Eason phát triển năm 1992. Các bước chính của mô hình SSM được thể hiện trên hình 5.1.



Hình 5.1. Mô hình hệ thống phần mềm.

Giai đoạn thứ nhất của mô hình SSM là nhận biết bài toán và khởi tạo bước phân tích. Điều này được tiếp tục bởi một mô tả chi tiết về tình huống bài toán, tạo nên một bức tranh phong phú cho phát triển, bao gồm tất cả các cổ động, các nhiệm vụ mà họ phải thực hiện, các nhóm họ tham gia.

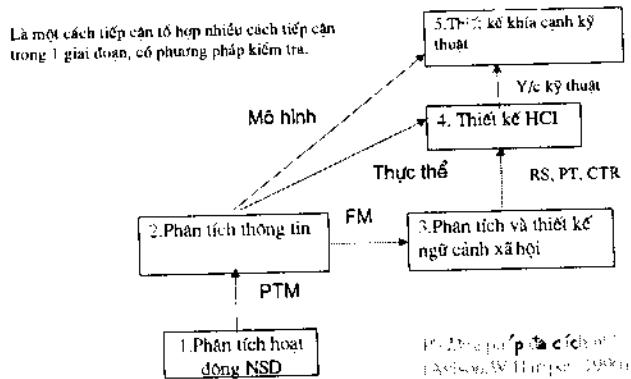
Ở bước tiếp theo, chúng ta chuyển từ thế giới thực sang hệ thống bằng cách cố gắng khởi tạo một định nghĩa gốc cho hệ thống.

SSM là một cách tiếp cận mềm dẻo, hỗ trợ việc xem xét một cách chi tiết ngữ cảnh của quá trình thiết kế. Tuy nhiên nó cần có thực tế để sử dụng hiệu quả. Nó không chỉ đơn giản là cung cấp câu trả lời đúng hay sai; việc sử dụng có kết quả nếu nó trợ giúp các nhà thiết kế hiểu hệ thống nhiều hơn.

### 5.2.3. Mô hình đa cách nhìn (Multiview)

Một cách tiếp cận có tính quy tắc là gắn hệ thống phần mềm và xã hội vào một phương pháp có tính hoàn thiện, đó chính là mô hình đa cách nhìn. Mô hình này do Avison và Wood-Harper đưa ra năm 1990. Mô hình này được thể hiện chi tiết trên hình 5.2.

## Chương 5: Các mô hình nhận thức người dùng



Hình 5.2. Mô hình đa cách nhìn (Avison & Wood - Harper).

Mô hình nhiệm vụ chủ yếu PTM tương tự như mô hình của Checkland trong SSM. Nó mô tả mục đích của hệ thống bao gồm các cổ động và chủ hệ thống. Bước thứ hai phân tích thông tin liên quan đến mô hình khái niệm của luồng dữ liệu và cấu trúc thông tin. Mô hình chức năng FM sử dụng ở giai đoạn ba như là cơ sở cho phân phối nhiệm vụ người dùng phân tích. Tập các vai trò RS và các nhiệm vụ yêu cầu máy tính CTR. Điều này chỉ được thực hiện sau các đặc tả kỹ thuật được làm.

Phương pháp này bao gồm cả cách tiếp cận kỹ thuật xã hội bao gồm cả các mô hình thực thể để biểu diễn cấu trúc thông tin. Mô hình này thể hiện nhiều ưu điểm. Thứ nhất nó chỉ ra các thành phần phải được làm trong quá trình thiết kế giao tiếp người dùng-máy tính. Thứ hai, nó cung cấp nhiều định hướng cho người thiết kế hệ thống. Nó cũng nhấn mạnh thứ tự các hoạt động phải tiến hành.

### 5.3. Mô hình nhận thức

Các mô hình này nhằm mục đích biểu diễn người dùng khi họ tương tác với giao diện, có nghĩa là nhằm mô hình hóa một số sắc thái của người dùng như: tri thức, chú ý hay xử lý. Các lớp mô hình này khác nhau từ các mô hình của nhiệm vụ ở mức độ cao và kết quả của các hoạt động giải quyết vấn đề đến việc mô tả các hành động ở mức thao tác, như mức vật lý nhấn phím trên bàn phím hay phím chuột.

Cách phân loại các loại mô hình này có thể dựa vào cách thức mô tả các đặc trưng về khả năng và hiệu năng của người dùng. Các mô hình hướng khả năng có khuynh hướng dự đoán một dãy các hành vi hợp lệ, cái hiện thời có thể thực hiện bởi người dùng. Ngược lại, mô hình hiệu năng không chỉ miêu tả một chuỗi hành vi cần thiết mà thường còn mô tả cái mà người dùng cần phải biết và cách thức sử dụng để thực hiện nhiệm vụ hiện tại. Trong các mô hình thuộc lớp này, chúng ta sẽ tập trung mô tả ba mô hình quan trọng:

- Mô hình GOMS nhằm biểu diễn cấu trúc đích của người dùng và phân cấp nhiệm vụ
- Mô hình ngôn ngữ và văn phạm
- Mô hình mức vật lý Keystroke.

### 5.3.1. Mô hình GOMS : Goals-Operator-Methods and Selection

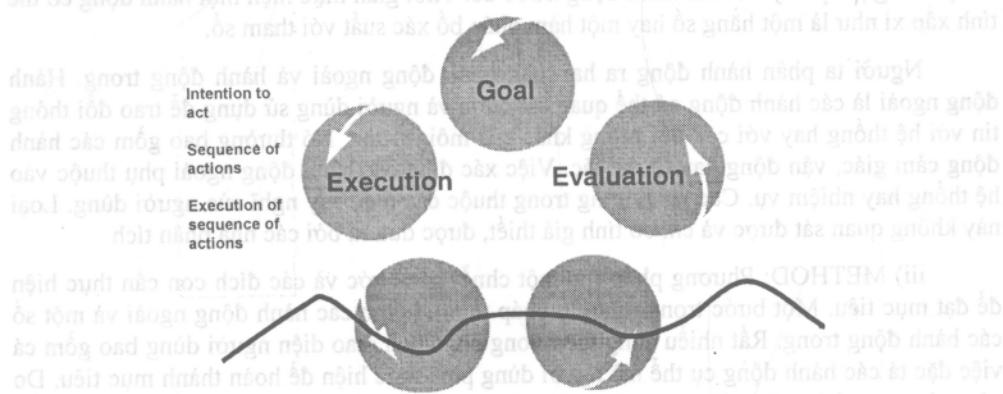
GOMS dựa vào nguyên tắc về tính hợp lý của mô hình bộ xử lý con người như điểm bắt đầu. Phương thức quan niệm của GOMS được đánh giá theo hai hướng: phân tích nhiệm vụ và lượng giá trước phản ứng của người dùng khi hoàn thành nhiệm vụ. GOMS viết tắt của Goals (mục đích), Operator (thao tác), Method (phương thức) và Selection Rules (luật lựa chọn). Nói một cách ngắn gọn, GOMS mô tả một tập các phương thức cần thiết để thực hiện mục đích. Phương thức bao gồm một dãy các thao tác mà người dùng có thể lựa chọn để thực hiện một đích con. Nếu có nhiều hơn một phương thức được chọn, GOMS sẽ cung cấp luật để chọn. Người ta coi mô hình GOMS gồm có bốn mô hình con sau:

- CMN-GOMS: được trình bày đầu tiên bởi Card, Moral và Newell. Đó là các định nghĩa còn lỏng lẻo về đích, đích con (sub-goal), các thao tác, các phương thức và các luật lựa chọn.

- KLM: một dạng đơn giản của CMN và được gọi là Keystroke Level Model. Nó chỉ sử dụng các thao tác nhấn phím, không có đích, phương thức và luật chọn. Nhà phân tích đơn giản liệt kê các phím nhấn và sự di chuyển chuột mà người dùng phải thực hiện để hoàn thành nhiệm vụ. Sau đó sử dụng các phương pháp tối ưu đơn giản để đưa ra các thao tác suy nghĩ.

- NGOMSL: là một dạng của GOMS được định nghĩa một cách chặt chẽ hơn, mô tả các thủ tục để xác định tất cả các thành phần của GOMS, thể hiện ở dạng giống như ngôn ngữ lập trình máy tính.

- CPM-GOMS: là dạng hành động song song sử dụng các thao tác kinh nghiệm (O), cảm giác (P) và vận động (M) trong biểu đồ PERT để có thể biết được hành động nào có thể hành động song song.



**Hình 5.3. Cấu trúc chu trình hành động con người của D. Norman và GOMS.**

GOMS là một mô hình hiệu năng theo nghĩa liên quan đến khía cạnh phương thức quan niệm của người dùng. Các phương thức mà GOMS đưa ra giống như cái mà các nhà tin học thực hiện. GOMS bắt đầu sau công việc phân tích nhiệm vụ: để áp dụng được kỹ thuật GOMS, nhà phân tích phải thực hiện quá trình phân tích nhiệm vụ để xác định được mục đích của người dùng.

## **Chương 5: Các mô hình nhận thức người dùng**

Nhà thiết kế có thể nhấn mạnh vào một mô hình GOMS để người dùng có thể thực hiện được mục đích trên hệ thống sẽ xây dựng. Tuy nhiên, GOMS không đưa ra lý thuyết để cấu trúc hoá nhiệm vụ, người dùng đòi hỏi phải có kinh nghiệm. GOMS cũng không nhầm vào một số kết quả của giao tiếp người dùng như: khả năng đọc, khả năng nhớ. GOMS cũng không nhầm vào tác động xã hội hay tổ chức. GOMS gồm 4 thành phần :

. i) GOAL: Mục tiêu - là tất cả những gì người dùng cần đạt được. Các mục tiêu lại thường được chia nhỏ thành các đích con. Và để đạt được mục tiêu chính, các mục tiêu con phải được thực hiện. Các mục tiêu thường được sắp xếp theo thứ tự phân cấp, song không bắt buộc.

Một mô tả mục đích là một cặp **hành động-đối tượng** theo kiểu <*động từ danh từ*>, thí dụ như *xoá từ* hay *di chuyển bằng bàn phím*,... Động từ có thể phức tạp nếu cần thiết phải phân biệt các phương thức.

ii) OPERATOR: Hành động mà người dùng dùng để thực hiện nhiệm vụ. Có một điểm khác biệt quan trọng giữa hành động và mục tiêu dù rằng cả hai đều có dạng **hành động-đối tượng**. Thí dụ *xem tài liệu* là của mục tiêu và nhấn phím là của **hành động**. Tuy nhiên, trong mô hình GOMS, mục tiêu là cái cần phải đạt được, trong khi đó hành động là cái dùng để thực hiện để đạt được mục tiêu. Do vậy, trong quá trình phân tích, hành động không phải chi tiết, còn mục tiêu phải được cung cấp thêm thông tin. Một cách đơn giản để phân biệt mục tiêu và hành động là hãy hỏi người dùng “bạn đang làm gì đây?” và ta sẽ có câu trả lời về mục tiêu chứ không phải là hành động, thí dụ kiểu “tôi đang làm cái này” chứ không phải là ‘tôi đang nhấn phím’.

Các thông số quan trọng của hành động là thời gian thực hiện được đánh giá độc lập với phương pháp hay với các hành động trước đó. Thời gian thực hiện một hành động có thể tính xấp xỉ như là một hằng số hay một hàm phân bố xác suất với tham số.

Người ta phân hành động ra hai loại: hành động ngoài và hành động trong. Hành động ngoài là các hành động có thể quan sát được và người dùng sử dụng để trao đổi thông tin với hệ thống hay với các đối tượng khác của môi trường. Nó thường bao gồm các hành động cảm giác, vận động hay tương tác. Việc xác định các hành động ngoài phụ thuộc vào hệ thống hay nhiệm vụ. Các hành động trong thuộc các mức suy nghĩ của người dùng. Loại này không quan sát được và chỉ có tính giả thiết, được đưa ra bởi các nhà phân tích

iii) METHOD: Phương pháp - là một chuỗi các bước và các đích con cần thực hiện để đạt mục tiêu. Một bước trong phương pháp thường gồm các hành động ngoài và một số các hành động trong. Rất nhiều công việc trong phân tích giao diện người dùng bao gồm cả việc đặc tả các hành động cụ thể mà người dùng phải thực hiện để hoàn thành mục tiêu. Do vậy, phương pháp cũng là trọng tâm của phân tích nhiệm vụ. Dạng mô tả phương pháp thường biểu diễn:

Method cho Goal : <Mô tả goal>

Bước 1: <hành động>

Bước 2: <hành động>

.....

Bước n: < kết thúc>

## Chương 5: Các mô hình nhận thức người dùng

Do có thể có nhiều hơn một hành động trong một bước, do đó phải dùng các phương pháp con để mô tả việc thực hiện các đích con.

iv) SELECTION RULE: Lật lựa chọn - là cách thức chọn lấy một phương pháp phù hợp để hoàn thành mục đích. Rõ ràng là nếu có nhiều hơn một phương pháp thì cần phải có luật chọn. Thí dụ: nếu mục đích là định vị con trỏ trên một văn bản thì ta có thể lựa chọn việc dùng chuột hay sử dụng bàn phím. Nếu từ vị trí hiện tại đến vị trí mới khá xa thì sử dụng chuột là hợp lý; ngược lại nếu khoảng cách là nhỏ ta sử dụng bàn phím. Dạng mô tả chung cho luật chọn dùng kiểu IF ... Then (Nếu ... thì):

Nếu <điều kiện> thì thực hiện đích <mô tả cho đích cụ thể>

Nếu <điều kiện> thì thực hiện đích <mô tả cho đích cụ thể>

### Kết thúc.

Ví dụ về di chuyển con trỏ trong soạn thảo văn bản :

Luật chọn cho mục đích: **di chuyển con trỏ**

IF đích hiện trên màn hình THEN Di chuyển bằng bàn phím lên/xuống

IF đích không hiện trên màn hình nhưng khoảng cách ngắn THEN

    Di chuyển bằng bàn phím cuộn

IF đích không hiện trên màn hình nhưng khoảng cách lớn và mô tả nhiệm vụ có chứa xâu cần tìm THEN

    Di chuyển bằng hàm

### Kết thúc

Để có ý tưởng đầy đủ về một mô tả GOMS, ta xét thí dụ về máy PHOTOCOPY như sau:

GOAL: PHOTOCOPY-PAPER

    GOAL: LOCATE-ARTICLE

    GOAL: PHOTOCOPY-PAGE lặp cho đến khi photo xong

        GOAL: ORIENT PAGE

            OPEN-COVER

            SELECT-PAGE

            POSITION-PAGE

            CLOSE-COVER

        GOAL: VERIFY COPY

            LOCAT-OUT-TRAY

            EXAMIN-COPY

        GOAL: COLLECT-COPY

            LOCATE-OUT-TRAY

            REMOVE-COPY (dịch ngoài được đáp ứng)

        GOAL: RETRIEVE-JOURNAL

            OPEN-COVER

            RETRIEVE-JOURNAL

            CLOSE-COVER

Đây chỉ là một lời giải với giả thiết: mỗi lần chỉ photo một trang và trang cần photo đã nằm sẵn trên mặt khay.

## *Chương 5: Các mô hình nhận thức người dùng*

### **Ưu nhược điểm của GOMS**

- GOMS chỉ là mô hình hiệu năng theo nghĩa liên quan đến khía cạnh phương thức của quan niệm và kỹ thuật đánh giá.

- GOMS đưa ra phương thức quan niệm giống như các nhà tin học thực hiện. Sự mô hình hoá một nhiệm vụ có thể được tinh chế hay biến đổi từ các thành phần tạo ra. Nó cung cấp cho các nhà thiết kế những hỗ trợ hình thức để đánh giá trước hiệu năng.

- GOMS không hỗ trợ lý thuyết để cấu trúc hoá nhiệm vụ, bản thân nó cũng quá giàn lược. Hơn nữa, GOMS chỉ ra quá trình thực hiện nhiệm vụ với giả thiết là không có lỗi. Song lỗi là điều khó tránh.

### **5.3.2. Mô hình ngôn ngữ**

Giao tiếp người dùng-máy tính thường được xem xét trong các thuật ngữ của một ngôn ngữ, do vậy không có gì ngạc nhiên khi nhiều mô hình hình thức được phát triển xung quanh khái niệm này. Trong số này, văn phạm BNF thường được dùng để đặc tả đối thoại, giúp ta hiểu được hành vi của người dùng và phân tích các khó khăn về nhận thức của tương tác. Dưới đây, chúng ta sẽ đề cập hai mô hình tiêu biểu: ký pháp văn phạm BNF (Backus-Naur-Form) và văn phạm nhiệm vụ - hành động TAG (Task - Action - Grammar).

#### **1) Ký pháp BNF**

BNF bao gồm các luật để mô tả đối thoại, chỉ liên quan đến cú pháp mà bỏ qua ngữ nghĩa của ngôn ngữ. Nó được dùng khá rộng rãi để đặc tả cú pháp của các ngôn ngữ lập trình máy tính. Thí dụ, xét một hệ thống đồ họa có chức năng “vẽ đường”. Để kích hoạt chức năng này, người dùng chọn mục vẽ đường trên menu. Chức năng này cho phép người dùng vẽ nhiều đoạn thẳng nối giữa 2 điểm. Người dùng chọn 1 điểm bằng cách nhấn chuột trong vùng vẽ và chỉ ra điểm cuối cùng bằng cách nhấn kép.

##### **Cú pháp**

vẽ đường ::= <chọn đường><chọn điểm><chọn điểm cuối>

chọn đường ::= <định vị con trỏ><NHÂN PHÍM ĐƠN>

chọn điểm ::= <chọn 1 điểm>/<chọn điểm><chọn 1 điểm>

chọn 1 điểm ::= <định vị con trỏ><NHÂN PHÍM ĐƠN>

chọn điểm cuối ::= <định vị con trỏ><NHÂN PHÍM KÉP>

định vị con trỏ ::= <rỗng><di chuyển><định vị con trỏ>

với quy ước:

- BNF dùng 2 loại ký hiệu: ký hiệu kết thúc và ký hiệu không kết thúc. Ký hiệu kết thúc viết bằng chữ in hoa, còn ký hiệu không kết thúc bằng chữ in thường.

- ::= hiểu là “được định nghĩa”

Một hạn chế của BNF là nó 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.

#### **2) Văn phạm nhiệm vụ-hành động**

Để khắc phục một số hạn chế của BNF như không dù tri thức nhận biết, văn phạm TAG thử đưa vào một số phần tử như luật văn phạm có tham số nhằm nhấn mạnh tính nhất

## **Chương 5: Các mô hình nhận thức người dùng**

---

quán và mã hoá tri thức của người dùng. Để tiện minh họa ta xét một số lệnh của UNIX: cp - sao tệp, mv - chuyển tệp và lk - liên kết. Mỗi lệnh đều có 2 dạng, phụ thuộc vào đối số là tệp hay thư mục:

- cp ::= ‘cp’ + filename + filename / ‘cp’ + filename + directory
- mv ::= ‘mv’ + filename + filename / ‘mv’ + filename + directory
- lk ::= ‘lk’ + filename + filename / ‘lk’ + filename + directory

Biện pháp dựa vào BNF rõ ràng không thể phân biệt được tính nhất quán của lệnh và một biến thể không nhất quán nếu “in” nhận đối số thứ nhất là thư mục. TAG được thiết kế nhằm khắc phục khuyết đó.

Mô tả lệnh trong văn phạm TAG có dạng sau (minh họa cho các lệnh trên):

```
File(Op) := command(Op) + filename + filename  
           /command(Op) + filename + directory  
command(op=cp) := ‘cp’  
command(op=mv) := ‘mv’  
command(op=lk) := ‘lk’
```

### **5.3.3. Mô hình Keystroke - KLM**

Mô hình KLM (Keystroke -Level Model) do Card, Moral và Newell đề xuất năm 1983 nhằm đánh giá thời gian thực hiện một nhiệm vụ từ một thiết kế xác định với kịch bản nhiệm vụ cụ thể. Về cơ bản, người ta liệt kê tuần tự các hành động ở mức nhấn phím (keystroke level) mà người dùng phải thực hiện để hoàn thành một công việc, sau đó đối chiếu thời gian cần thiết để thực hiện công việc. Ở đây, không nhất thiết phải có thiết kế hay chế thử, KLM chỉ yêu cầu giao diện người dùng phải được mô tả đủ chi tiết để có thể thực hiện tuần tự các hành động được yêu cầu để hoàn thành nhiệm vụ mong muốn.

Các hành động được gọi ở mức nhấn phím nếu chúng có cùng mức với các hành động như nhấn phím, di chuột, nhấn phím chuột. Mô hình KLM 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. Có một tập các oprator chuẩn được sử dụng trong mô hình KLM mà thời gian thực hiện chung được lượng giá theo các dữ liệu thực nghiệm.

KLM biểu diễn tương tác giữa người và máy ở dạng một ngôn ngữ có cú pháp và từ vựng. Những thành phần của nó là các hành động vật lý mà người dùng phải thực hiện để hoàn tất nhiệm vụ.

Việc áp dụng mô hình KLM để lượng giá thời gian thực hiện một nhiệm vụ bao gồm các bước sau:

- i) Chọn một hay nhiều kịch bản nhiệm vụ tiêu biểu
- ii) 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.

- iii) Với mỗi kích bản nhiệm vụ hãy tìm ra cách tốt nhất để hoàn thành hoặc cách mà bạn nghĩ người dùng sẽ sử dụng.
- iv) Liệt kê các hành động mức phím nhấn hay mức vật lý tương ứng với việc thực hiện nhiệm vụ.
- v) Thêm các hành động suy nghĩ khi người dùng dừng lại hay suy nghĩ.
- vi) Xác định thời gian thực hiện chuẩn cho mỗi hành động này.
- vii) Nếu cần có thể thêm các hành động mà người dùng phải chờ sự đáp ứng của hệ thống.
- viii) Thời gian thực hiện nhiệm vụ được lượng giá bằng tổng thời gian thực hiện của tất cả các hành động.

Mô hình được áp dụng với các giả thiết:

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

#### Các thao tác và nguyên tắc tính thời gian thực hiện

KLM đưa ra sáu thao tác để mô tả việc thực hiện một nhiệm vụ cơ bản:

- a) K-Kestrocking: Gõ phím hay nhấn chuột.
- b) P-Pointing: Di chuyển con trỏ chuột tới một điểm trên màn hình.
- c) H-Homing: Thay đổi công cụ.
- d) D-Drawing: Hành động vẽ.
- e) M (Mental activity): Hành động suy nghĩ.
- f) R (Response Time): Thời gian đáp ứng của hệ thống.

Nếu gọi thời gian cần thiết cho các thao tác cơ bản trên là  $T_K$ ,  $T_P$ ,  $T_H$ ,  $T_D$ ,  $T_M$  và  $T_R$  thì tổng thời gian thực hiện một thao tác sẽ là:

$$T_{exc} = T_K + T_P + T_H + T_D + T_M + T_R$$

- $K$  là hành động nhấn một phím trên bàn phím hay phím chuột. Thời gian thực hiện hành động này từ 0,28 s đến 1,2 s phụ thuộc vào kinh nghiệm của người dùng. Thị dụ người đánh máy chuyên nghiệp (90 từ/phút) thời gian này là 0,12 s, người không quen bàn phím là 1,2 s. Theo các chuyên gia,  $T_K$  được tính xấp xỉ như sau:

$$T_K = (\text{tổng thời gian thử/ số phím nhấn})$$

Từ đây, dễ dàng tính được  $T(n)$  - thời gian gõ lần lượt  $n$  phím:  $T(n) = n \times T_K$

## Chương 5: Các mô hình nhận thức người dùng

- P là hành động di chuyển con trỏ tới 1 điểm trên màn hình với thời gian là 1,1 s. Thời gian này có thể tính theo luật Fitts:  $T_p = (K_0 \cdot T_K) + I \log_2(D/L) + 0,5$ . Thường thời gian thực hiện hành động này từ 0,8 đến 1,5 s. Nếu không cần tính chính xác cao, ta có thể sử dụng thời gian trung bình là 1,1s.
- D là hành động vẽ dùng chuột.  $T_D$  phụ thuộc tuyến tính vào số lượng nét vẽ và độ dài nét vẽ.  $T_D$  được tính:  $T_D = 0,9n + 0,161$ .
- H là hành động thay đổi công cụ (từ bàn phím sang chuột hoặc ngược lại). Thời gian  $T_H$  được tính từ khi chuyển từ công cụ này cho đến khi nắm chắc công cụ mới. Do việc chuyển này là khá nhanh và theo kinh nghiệm,  $T_H$  là 0,4 s.
- M là hoạt động tinh thần của suy nghĩ hay nhận thức. Đương nhiên hành động suy nghĩ phụ thuộc vào quá trình nhận thức và khác nhau đối với nhận thức khác nhau và với các cá thể khác nhau. Hành động này dựa trên thực tế là khi người dùng có kinh nghiệm đang thực hiện các hành động với máy tính, có sự tạm dừng trong chuỗi hành động ngăn người để nhớ lại một cái gì đó, thí dụ tên tệp hoặc tìm một đối tượng trên mô hình. Hành động M có xu hướng mô tả các suy nghĩ thông thường, không phức tạp hoặc quá dài và thường có sự đồng đều. Dựa trên các kết quả có sẵn (Olson & Olson, 1990), thời gian lượng giá cho một hành động M là khoảng 1,2 s (từ 0,6 đến 1,35 s). Việc lựa chọn xem có bao nhiêu hành động M và xảy ra ở đâu là điều khó nhất khi áp dụng mô hình KLM. Một số luật Heuristic đã được xem xét cho vấn đề này và sẽ được giới thiệu ở phía dưới.
- R là hành động chờ đợi sự đáp ứng của hệ thống trước khi người dùng có thể tiếp tục chuỗi hành động của mình. Chú ý rằng nó không cần phải giống thời gian hệ thống yêu cầu vì người dùng có thể xếp chồng các thao tác khi hệ thống đang làm việc. Với các máy tính hiện đại, thời gian chờ đợi này là không đáng kể. Trong những trường hợp khác như sự phục hồi dữ liệu, sự chậm trễ này có thể là quan trọng. Hành động R có thể hiểu rộng hơn là hành động W (Waiting time) diễn tả hành động mà người dùng phải chờ chứ không hoàn toàn là thời gian đáp ứng của hệ thống.

Ngoài các hành động chính trên, ta có thể bổ sung thêm hai hành động phụ:

- B là hành động nhấn đơn trên phím chuột. Việc nhấn phím/ nhả phím diễn ra khá nhanh (0,1 s);
- BB là hành động nhấn kép và chiếm khoảng 0,2 s.

### Một số luật Heuristic

M phụ thuộc vào các hành động vật lý và khả năng của người dùng, là tham số với người dùng. Một số luật Heuristic sau có thể được áp dụng để thêm vào hay loại bỏ đi một số M:

- **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. Thí dụ trong dãy PMK, với K là hành động nhấn phím chuột thì

## *Chương 5: Các mô hình nhận thức người dùng*

hành động tâm lý cho K có thể tính trước khi đang ở trong P.

- **Luật 2:** Nếu dây 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 thì loại bỏ M trước K (một ký tự kết thúc rườm rà nếu nó theo sau một ký hiệu báo sự kết thúc khác).
- **Luật 4:** Nếu K xác định một hằng số thì bỏ M, ngược lại khi K xác định một biến số thì M cần được giữ.

### **Một số thí dụ**

**Thí dụ 1:** Thời gian thực hiện thao tác bôi một đoạn văn bản dùng chuột

#### 1) Dãy các hành động

- Nắm lấy chuột;
- Định vị vào vị trí cần bôi đen;
- Nhấn và giữ phím chuột;
- Kết thúc bằng việc nhả phím;
- Định vị con trỏ vào biểu tượng B trên thanh công cụ;
- Nhấn phím chuột;
- Chuyển lại bàn phím.

#### 2) Mã hoá

H(chuột)P(chuột)K(nút chuột) K(nút chuột) P(chuột)K(nút chuột)H(bàn phím)

#### 3) Áp dụng các luật

**Luật 0:** thêm các thao tác M trước K và P

H(chuột) MP(chuột) MK(nút chuột) MK(nút chuột) MP(chuột) MK(nút chuột)  
H(bàn phím)

**Luật 1:** PMK => PK

H(chuột) MP(chuột) K(nút chuột) MK(nút chuột) MP(chuột) K(nút chuột) H(bàn phím)

Như vậy, thời gian tổng cộng là  $2(H + P) + 3(M + K)$ . Thay các giá trị T tương ứng ta sẽ có tổng thời gian thực hiện nhiệm vụ này.

**Thí dụ 2:** Thời gian thực hiện thao tác xoá tệp bằng phương pháp kéo thả và người dùng mới

#### 1) Dãy các hành động

- Quyết định thực hiện công việc: hành động M
- Tìm biểu tượng tệp cần xoá: hành động M
- Trỏ vào biểu tượng tệp: hành động P
- Nhấn và giữ phím chuột: hành động K
- Kiểm tra biểu tượng có thay đổi: hành động M

## **Chương 5: Các mô hình nhận thức người dùng**

---

- Tìm biểu tượng thùng rác: hành động M
- Thả tệp vào thùng rác: hành động P
- Kiểm tra thùng rác có thay đổi: hành động M
- Nhá phím chuột: hành động K
- Kiểm tra thùng rác phình ra: hành động M
- Tìm cửa sổ ban đầu: hành động M
- Trò vào cửa sổ ban đầu: hành động P.

### **2) Mã hoá**

Việc mã hoá là khá đơn giản và coi như bài tập tự làm.

### **3) Áp dụng các luật**

Sau khi áp dụng các luật, chúng ta dễ dàng thấy thời gian tổng cộng là:

$$3P + 2K + 7M = 13.2s$$

**Thí dụ 3:** Thời gian thực hiện thao tác xoá tệp bằng phương pháp kéo thả và người dùng có kinh nghiệm.

Dãy các hành động

- Quyết định thực hiện công việc: hành động M
- Tìm biểu tượng tệp cần xoá: hành động M
- Trò vào biểu tượng tệp: hành động P
- Nhấn và giữ phím chuột: hành động K
- Thả tệp vào thùng rác: hành động P
- Nhá phím chuột: hành động K
- Trò vào cửa sổ ban đầu: hành động P.

Thời gian tổng cộng là  $3P + 2K + 2M = 6,5 s$ .

Điều này là hợp lý vì với người dùng có kinh nghiệm, họ sẽ không tốn thời gian vào các hành động suy nghĩ, thời gian thực hiện nhiệm vụ sẽ giảm đi.

### **Một số ưu nhược điểm của mô hình KLM**

KLM là một công cụ phân tích định lượng. Nó cho phép so sánh và tiên đoán hiệu năng với người dùng trong một cú pháp cho trước. Thi dụ KLM cho phép khẳng định hiện cả trang nhanh hơn hiện từng dòng, chọn chữ bằng chuột nhanh hơn dùng bàn phím. Một ưu điểm nữa của KLM là đơn giản, hệ thống thuật ngữ và kỹ thuật mô hình khá dễ hiểu với người dùng không chuyên trong lĩnh vực tâm lý học nhận thức.

Tuy nhiên KLM cũng bộc lộ một số hạn chế. Thứ nhất, việc thay các M là không đơn giản. Thứ hai là sự không cụ thể của việc lượng giá vì trong mô hình không tính đến thời gian đáp ứng của hệ thống mà đây cũng là một yếu tố quan trọng. Thứ ba, tuy rằng có đề ý đến phím song nó không có sự phân biệt cụ thể giữa các phím với nhau. Thực tế đã chứng minh là có sự khác nhau nhiều giữa nhá phím thông dụng với các phím đặc biệt. Điều cuối cùng, KLM chỉ quan tâm đến thời gian thực hiện chứ không quan tâm đến tổng thời gian hoàn thành nhiệm vụ.

**Tài liệu tham khảo**

- [1] *Alain J. Dix, Finlay, Abowd and Beale. Human-Computer Interaction (course and book)*, Prentice-Hall, second edition, 1993.
- [2] *J. Coutas. Interface Home-ordinateur, Conception et réalisation*, Dunod Paris 1990.
- [3] *Bonnie E. John, David E. Kieras. The GOMS Family of User Interface Analysis Techniques: Comparaison and Constrat.*
- [4] *David E. Kieras. Using the Keystroke - Level Model to Estimate Execution Times.*
- [5] [www.ei.es.vt.edu](http://www.ei.es.vt.edu).

# 6

## ĐẶC TẢ YÊU CẦU VÀ PHÂN TÍCH NHIỆM VỤ

### Nội dung

- 6.1. Đặc tả nhu cầu người dùng
  - 6.1.1. Đặc tả chức năng
  - 6.1.2. Đặc tả dữ liệu
  - 6.1.3. Đặc tả tính dùng được
- 6.2. Công cụ
- 6.3. Phân tích nhiệm vụ (Task Analysis - TA)
  - 6.3.1. Vai trò
  - 6.3.2. Một số thuật ngữ: Goal, Task, Action
  - 6.3.3. Phân tích nhiệm vụ phân cấp (HTA  
Hierachical Task Analysis)
  - 6.3.4. Phân tích nhiệm vụ theo *nhan thức*
  - 6.3.5. Phân tích nhiệm vụ theo *mô hình tri thức*  
“How-to-do-it”
- 6.4. Thi dụ: Quản lý tệp của PC-DOS và Macintosh

Đặc tả yêu cầu người dùng và phân tích nhiệm vụ là hai giai đoạn quan trọng trong phát triển giao tiếp người dùng-máy tính. Đặc tả yêu cầu người dùng là quá trình xác định các nhu cầu của người dùng với hệ thống mà chúng ta sẽ phát triển. Nó là kết quả của quá trình thu thập yêu cầu người dùng hay phân tích hệ thống và thường đặc tả bằng các công cụ đồ họa hay các tài liệu hình thức.

**Phân tích nhiệm vụ (TA)** là quá trình phân tích cách thức người dùng thực hiện công việc để đạt được mục đích của mình. Phân tích này tập trung vào các hành động của người dùng (actions), các đối tượng mà người dùng tác động vào (objects), và các hiểu biết mà người dùng cần có (knowledge).

### 6.1. Đặc tả nhu cầu người dùng

Với mọi hệ thống phần mềm, việc thu thập các yêu cầu người dùng là công việc đầu tiên. Thông thường, khách hàng cung cấp một bản mô tả về những yêu cầu của mình, cái mà họ mong muốn hệ thống mới phải cung cấp trong thuật ngữ của họ. Nói chung mô tả này còn khá chung chung và cần phải tìm hiểu làm rõ và xác định những cái không thể thực hiện được, những cái còn thiếu, nhập nhằng hoặc không chắc chắn. Nói chung, các kỹ thuật sử dụng trong giai đoạn này là phòng vấn, quan sát và phân tích tài liệu. Nhà phân tích có thể phân tích nhiệm vụ của người dùng bằng các kỹ thuật của khoa học nhận thức hay các kỹ thuật phân tích nhiệm vụ.

Thu thập yêu cầu người dùng không chỉ liên quan đến việc hiểu các yêu cầu mà còn trợ giúp các nhà thiết kế phân tích các tình huống. Kết quả của quá trình thu thập dữ liệu là:

- Một biểu diễn bài toán với hệ thống hiện tại.
- Một biểu diễn các yêu cầu của hệ thống mới.

Nhìn chung, các đặc tả yêu cầu của một hệ thống thường phân làm ba loại:

- Đặc tả chức năng;
- Đặc tả dữ liệu;
- Đặc tả tính dùng được (đây là đặc tả mới chỉ xuất hiện trong phát triển hệ thống giao tiếp người dùng-máy tính).

#### 6.1.1. Đặc tả chức năng

Đặc tả chức năng là đặc tả cái mà hệ thống phải làm. Ở đây hệ thống bao hàm cả máy tính và người dùng con người. Việc quyết định hoạt động nào thực hiện bởi máy tính, hoạt động nào thực hiện bởi con người được trì hoãn cho đến sau quá trình phân tích nhiệm vụ và với nhiều thiết kế khác phải được làm. Đặc tả chức năng cũng bao hàm cả việc đặc tả các ràng buộc mà các chức năng khi thực hiện phải tinh đến.

Việc đặc tả này thường được phân chia thành nhiều módun, được tổ chức theo kiểu phân cấp. Sự phân cấp này cho phép việc mô tả hệ thống phải được chia thành nhiều phần dễ điều khiển và có thể xử lý một cách riêng biệt. Mức cao nhất của sự phân cấp là mức khung cảnh - mức trừu tượng nhất. Mức cuối cùng là mức cụ thể nhất, đặc tả cái mà nó mong muốn thực hiện.

## **Chương 6: Đặc tả yêu cầu và phân tích nhiệm vụ**

---

Điều quan trọng cần biết là việc thu thập các yêu cầu chức năng không thể được làm và đặc tả chỉ trong một lần. Để đặc tả chức năng, người ta hay dùng các lưu đồ như biểu đồ luồng (DFD - Data Flow Diagram) bên cạnh các ngôn ngữ tự nhiên hay ngôn ngữ có cấu trúc.

### **6.1.2. Đặc tả dữ liệu**

Đặc tả dữ liệu bao gồm việc biểu diễn luồng dữ liệu, ngữ nghĩa và cấu trúc của chính dữ liệu đáp ứng yêu cầu của người dùng. Ngày nay, trong phần lớn các phương pháp phát triển hệ thống thông tin, người ta đều nhận thấy tầm quan trọng của việc đặc tả dữ liệu. Việc đặc tả dữ liệu ngược với đặc tả chức năng, nó chỉ tập trung vào cấu trúc dữ liệu.

Việc đặc tả dữ liệu được liệt kê nhờ các kỹ thuật quan sát, phân tích tài liệu, phỏng vấn,... Điều sống còn trong việc này là phải hiểu các phần tử dữ liệu, định nghĩa một cách chính xác và nó cần hiện diện trong hệ thống.

Trong quá trình đặc tả dữ liệu, từ điển dữ liệu là một công cụ không thể thiếu. Nó giúp nhà phân tích xác định ngữ nghĩa dữ liệu, nhất là các phần tử dữ liệu. Việc đặc tả dữ liệu cũng cần xác định mối quan hệ giữa các thực thể - cái cần cho ứng dụng và các ràng buộc trên các đối tượng này.

Phân tích dữ liệu liên quan tới việc thiết lập một cách chính xác các dữ liệu mà hệ thống yêu cầu, dữ liệu được cấu trúc ra sao và được lưu trữ một cách lôgic thế nào. Công cụ quan trọng trong đặc tả và phân tích dữ liệu là lược đồ dữ liệu - ERD (Entity Relation Diagram).

### **6.1.3. Đặc tả tính tiện dùng**

Trong Công nghệ phần mềm và trong Hệ thống thông tin, chúng ta mới chỉ quan tới đặc tả chức năng và đặc tả dữ liệu. Tuy nhiên, một sắc thái khác cũng cần được quan tâm trong giao tiếp người dùng-máy tính đó là đặc tả tính tiện dùng.

Tính tiện dùng theo chuẩn ISO 9241-11 được nêu trong chương mở đầu. Việc đặc tả tính tiện dùng được tiến hành đồng thời với đặc tả chức năng và đặc tả dữ liệu với cùng một số kỹ thuật. Hoạt động thu thập yêu cầu về tính tiện dùng thường được biệt đền với thuật ngữ “nghiên cứu tính tiện dùng” và có liên quan đến đánh giá quá trình. Việc xác định các yêu cầu của tính tiện dụng đòi hỏi người phân tích phải hiểu 3 kiểu chính của phân tích sẽ đề cập trong phần phân tích.

## **6.2. Công cụ**

Đặc tả yêu cầu là một giai đoạn rất quan trọng trong vòng đời phát triển phần mềm và có ảnh hưởng đến mọi giai đoạn trong vòng đời phần mềm. Do vậy cần phải có công cụ hỗ trợ cho quá trình đặc tả này. Công nghệ hệ thống thông tin đã sử dụng và mô tả nhiều công cụ hữu hiệu. Trong đặc tả chức năng, người ta có thể sử dụng sơ đồ luồng dữ liệu - DFD, máy với trạng thái hữu hạn - FST cùng với ngôn ngữ tự nhiên có cấu trúc. Trong đặc tả dữ liệu có lưu đồ thực thể liên kết - ER. Các công cụ này đã được mô tả khá chi tiết trong hai giáo trình trên và dễ dàng tìm thấy trong nhiều tài liệu. Vì vậy, để trong sáng cách trình bày, chúng tôi không trình bày chi tiết ở đây.

### 6.3. Phân tích nhiệm vụ

#### 6.3.1. Vai trò

Phân tích nhiệm vụ là một quá trình nhằm tìm hiểu cách thức mà người dùng con người hiểu công việc hay cái đích phải thực hiện, các đối tượng mà người dùng sẽ thao tác trên đó và những tri thức họ cần có để thực thi nhiệm vụ đạt đích mong muốn. Để hiểu rõ hơn vai trò của phân tích nhiệm vụ, xét ví dụ nhỏ sau: “Lau nhà”.

**Mục đích:** Lau nhà

**Các công việc cần làm:** - Lấy máy hút bụi

- Lắp các phụ tùng cần thiết

- Lau nhà

**Khi hộp rác đầy tháo bỏ rác và lắp lại**

**Khi lau xong tháo cất máy hút bụi**

Rõ ràng là để thực hiện các công việc trên, chúng ta cần có tri thức về việc sử dụng máy hút bụi như tháo hay lắp các chi tiết, trình tự lau các phòng,...

Tuy nhiên, một khó khăn mà chúng ta thường gặp phải là việc sử dụng thuật ngữ. Nhiều tác giả sử dụng cùng một thuật ngữ cho những cái khác nhau như: đích, nhiệm vụ và hành động hay thao tác, kế hoạch. Trước khi xem xét chi tiết các kỹ thuật áp dụng trong phân tích nhiệm vụ, cũng hữu ích khi làm rõ các khái niệm này. Tiếp theo, chúng ta sẽ tập trung đi vào một số các kỹ thuật chính trong phân tích nhiệm vụ như: Kỹ thuật phân rã nhiệm vụ - HTA, phân tích dựa vào tri thức và phân tích dựa vào nhận thức.

#### 6.3.2. Một số thuật ngữ: Mục đích (Goal), nhiệm vụ (Task) và hành động (Action)

##### Mục đích

Mục đích hay còn gọi là nhiệm vụ ngoài có thể được định nghĩa như một trạng thái của một hệ thống mà người dùng muốn hoàn thành [1]. Một cái đích có thể mô tả ở một mức độ trừu tượng đặc biệt.

Một đích có thể được thực hiện bởi một số các công cụ, phương pháp, tác nhân, kỹ thuật và có thể gọi là các thiết bị, cái có thể làm thay đổi trạng thái của hệ thống. Thí dụ để viết một bức thư, ta có thể dùng máy chữ, một hệ soạn thảo văn bản nào đó hay một cái bút và giấy. Một đích có thể gồm nhiều đích con hay nhiệm vụ (nhiệm vụ trong).

##### Nhiệm vụ

Một nhiệm vụ là một tập có cấu trúc các hoạt động, trong đó các hành động được làm theo một trình tự nào đó. Như vậy, nhiệm vụ là cái người dùng phải làm (hay họ cho là phải làm) nhằm thực hiện mục đích đặt ra. Tại một số thời điểm, con người tương tác vật lý với thiết bị bằng cách thực hiện một hành động hay một thao tác (operation). Thí dụ, để đưa vào một lệnh từ bàn phím, người dùng phải nhấn phím hoặc sử dụng thiết bị chuột.

##### Hành động

Một hành động [Card et al, 1983] định nghĩa là một nhiệm vụ mà bản thân nó không bao hàm việc giải quyết vấn đề hay là một thành phần của cấu trúc điều khiển. Hình 6.1 minh họa ý nghĩa của đích, nhiệm vụ và hành động.

## **Chương 6: Đặc tả yêu cầu và phân tích nhiệm vụ**

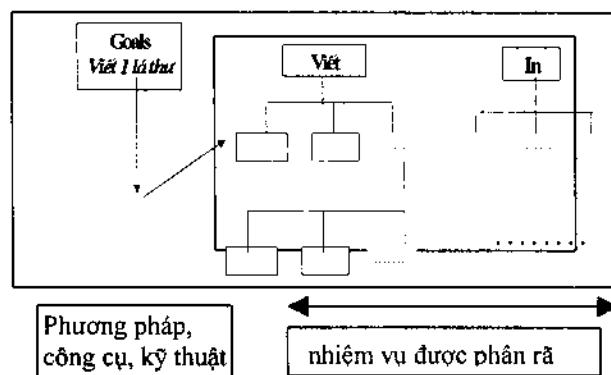
**Mục đích:** Viết một lá thư

**Thiết bị:** Phương pháp, công cụ, hoặc kỹ thuật.

**Nhiệm vụ:** Các hoạt động cần thiết để hoàn thành mục đích bằng các thiết bị.

**Nhiệm vụ con:** các thành phần của nhiệm vụ.

**Hành động:** là các nhiệm vụ đơn giản không có cấu trúc điều khiển.



**Hình 6.1. Mô hình khái quát cho phân tích nhiệm vụ.**

### **6.3.3. Kỹ thuật phân tích nhiệm vụ phân cấp (HTA Hierachical Task Analysis)**

Phần lớn các kỹ thuật phân tích nhiệm vụ đều sử dụng một số dạng của phân rã nhiệm vụ để biểu diễn các hành vi. Phân tích nhiệm vụ phân cấp là một cách tiếp cận như vậy với 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. Kỹ thuật này có từ hơn 20 năm và là một dạng nổi tiếng nhất của phân tích nhiệm vụ.

Xét thí dụ "Lau nhà" ở đầu chương và áp dụng việc phân rã cũng như kế hoạch thực hiện, ta được bảng sau:

#### 0 Mục đích lau nhà

1. Lấy máy hút bụi
2. Lắp các phụ tùng
3. Lau nhà
  - 3.1. Lau hành lang
  - 3.2. Lau phòng khách
  - 3.3. Lau phòng ngủ
4. Đổ túi chứa bụi
5. Tháo và cất máy

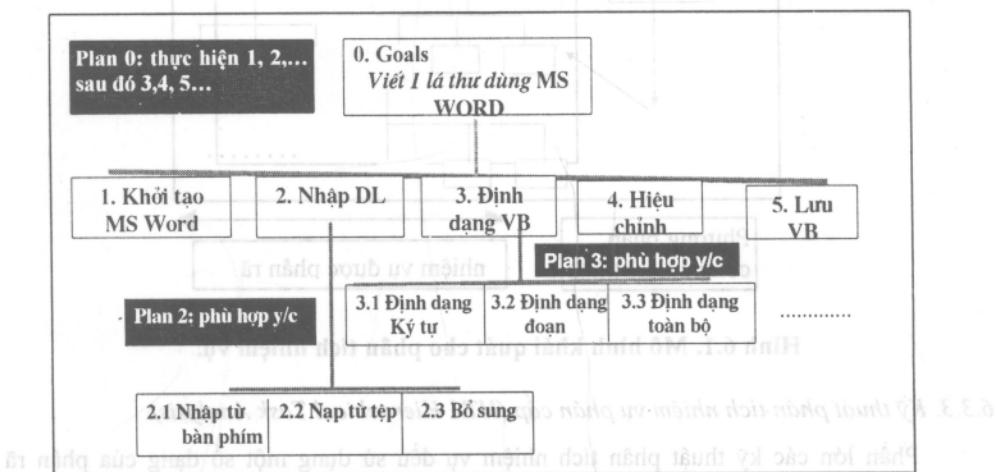
Plan 0: thực hiện các công việc 1-2-3-5 theo trình tự và khi túi chứa bụi đầy thực hiện 4.

Plan 3: Thực hiện các nhiệm vụ 3.1, 3.2, 3.3 theo thứ tự bất kỳ, phụ thuộc vào nhu cầu.

## Chương 6: Đặc tả yêu cầu và phân tích nhiệm vụ

Trong cách biểu diễn này người ta thường dùng ký pháp viết thực để biểu diễn các mức và đánh số các nhiệm vụ để nhấn mạnh tính phân cấp. Do vậy, kế hoạch 0 thể hiện cách chúng ta thực hiện các nhiệm vụ từ 1 đến 5 của đích 0. Tương tự, kế hoạch 3 biểu diễn cách thức thực hiện nhiệm vụ từ 3.1 đến 3.3, không có kế hoạch cho thực hiện các nhiệm vụ 1,2 và 4 vì nó không cần phân rã.

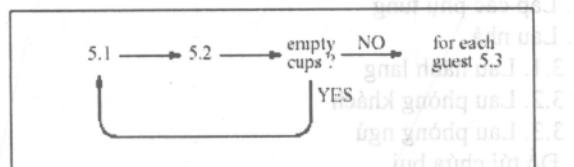
Ngoài ký pháp viết thực, người ta có thể dùng ký pháp đồ họa (hình 6.2). Các nhiệm vụ được đánh số để phân biệt bắt đầu từ 1 vì tự nó không nói lên thứ tự thực hiện. (nhiệm vụ số 0 được dành để biểu diễn mục đích). Các kế hoạch mới thể hiện trình tự. Các kế hoạch cũng được đánh số bắt đầu từ 0 và số hiệu thường gắn với các nhiệm vụ.



Hình 6.2. Cấu trúc HTA cho việc soạn thảo một lá thư bằng WORD.

Vì quá trình thực hiện nhiệm vụ có thể phụ thuộc điều kiện, do vậy cần biểu diễn nó dưới dạng rẽ nhánh, lặp và chờ đợi. thí dụ:

- chuỗi cố định: 1.1{1.2{1.3}
- lựa chọn: dùng luật nếu thì, thí dụ if the pot is full 2
- đợi sự kiện: thí dụ when kettle boils do 1.4
- chu trình:



- thời gian phân chia: **do 1; at the same time do ...**

Tuy nhiên, trong cách biểu diễn này cần bổ sung thêm luật kết thúc, vì nó không chỉ rõ khi nào thì đích được hoàn thành.

## ***Chương 6: Đặc tả yêu cầu và phân tích nhiệm vụ***

### ***6.3.4. Kỹ thuật phân tích nhiệm vụ theo nhận thức***

Trong khi HTA liên quan đến việc thiết lập cách biểu diễn chính xác của các bước phải làm nhằm thực hiện nhiệm vụ thì phân tích nhiệm vụ theo nhận thức lại tập trung vào sự biểu diễn các tri thức mà người dùng có hoặc cần phải có để hoàn thành nhiệm vụ. Phân tích nhiệm vụ nhận thức liên quan đến việc cung cấp cho quá trình thiết kế xuyên suốt ứng dụng về lý thuyết nhận thức. Chúng ta đã định nghĩa một nhiệm vụ như là cái mà con người hay các tác nhân phải làm hay tin là phải làm để hoàn thành mục đích bằng cách sử dụng một số thiết bị. Phân tích nhiệm vụ nhận thức nhận thấy rằng một số hành động là hoàn toàn vật lý như nhấn phím, di chuyển con trỏ, một số khác lại có tính suy nghĩ hoặc các thao tác nhận thức, thí dụ như quyết định phím nào sẽ được nhấn, con trỏ phải chuyển đến vị trí nào trên màn hình.

Trách nhiệm chủ yếu của tâm lý nhận thức là ở chỗ con người cảm nhận thế giới và tạo nên một số biểu diễn trong suy nghĩ của mình. Điều này đôi khi gọi là không gian bài toán. Sự biểu diễn này chúng ta thường gọi là “tri thức”. Nó được miêu tả bằng các thuật ngữ của các khái niệm mà chúng ta có, mối liên hệ giữa các khái niệm này và khả năng của chúng ta có thể sử dụng các khái niệm này. Khi đã có cách biểu diễn, con người sẽ điều khiển nó và tạo nên một số hành vi có thể quan sát được. Bảng dưới đây sẽ chỉ ra mối quan hệ giữa một số kỹ thuật phân tích nhiệm vụ nhận thức và các mô hình chung.

Kỹ thuật	Mục đích	Mức nhiệm vụ	Mức hành động
GOMS	Đích	Đích con	Thao tác, phương pháp
TAG		Nhiệm vụ	Hành động
ETIT	Nhiệm vụ ngoài	Nhiệm vụ trong	
YSS	Không gian bài toán	Không gian thiết bị	
CLG	Mức nhiệm vụ	Mức ngữ nghĩa	Mức độ cú pháp và từ vựng
KAT/TKS	Đích/cấu trúc đích con	cấu trúc nhiệm vụ	Hành động, thủ tục

với: ETIT - External Task Internal Task

KAT/TKS - Knowledge Analysis of Tasks/ Task Analysis Structure.

### ***6.3.5. Kỹ thuật phân tích nhiệm vụ theo tri thức - mô hình How to do***

Phân tích nhiệm vụ dựa vào tri thức bắt đầu bằng việc liệt kê tất cả các đối tượng và hành động liên quan tới nhiệm vụ và sau đó phân loại chúng. Điều này cũng giống như kỹ thuật HTA. Để dễ theo dõi, chúng ta bắt đầu bằng việc xem xét một thí dụ về phân cấp đối tượng đơn giản, đó là việc điều khiển động cơ ôtô. Việc liệt kê và phân cấp các đối tượng và các hành động có thể minh họa như sau:

## **Chương 6: Đặc tả yêu cầu và phân tích nhiệm vụ**

### **Điều khiển mô tơ**

Thiết bị lái	tay lái, đèn chỉ thị
Động cơ/hộp số	
Điều khiển	bộ phận đánh lửa, cần gạt số
Hệ thống bánh răng	bộ ly hợp,
Đèn	
Đèn ngoài	
Đèn trong	
Gạt nước	
Buồng đốt	
Bộ phận an toàn khi đỗ	
Đài	

### **6.4. Thí dụ**

Xét thí dụ "**Quản lý tệp**" trên PC-MSDOS và trên máy Macintosh, với các mục đích sau:

- Xoá tệp
- Chuyển tệp
- Xoá thư mục
- Chuyển thư mục

Trước hết, chúng ta biểu diễn dưới dạng văn bản theo ký pháp HTA. Tiếp sau, ta biểu diễn dưới dạng lưu đồ.

#### **A. Với máy PC**

##### **A.1. Các phương pháp quản lý tệp - 1**

###### **1. Phương pháp thực hiện mục đích xoá 1 tệp**

- Step 1 Chọn lệnh ERASE từ tệp lệnh
- Step 2 Nghĩ tên TM và tên tệp
- Step 3 Nhập lệnh và thực hiện lệnh
- Step 4 Kết thúc

###### **2. Phương pháp thực hiện mục đích chuyển 1 tệp**

- Step 1 Thực hiện sao 1 tệp
- Step 2 Thực hiện xoá 1 tệp
- Step 3 Kết thúc

## *Chương 6: Đặc tả yêu cầu và phân tích nhiệm vụ*

---

### *3. Phương pháp thực hiện mục đích sao 1 tệp*

- Step 1 Chọn lệnh COPY từ tệp lệnh
- Step 2 Nghĩ tên TM và tên tệp nguồn
- Step 3 Nghĩ tên TM và tên tệp đích
- Step 4 Nhập lệnh và thực hiện lệnh
- Step 5 Kết thúc

### **A.2. Các phương pháp quản lý tệp - 2**

#### *1. Phương pháp thực hiện mục đích xoá 1 thư mục*

- Step 1 Xoá tất cả các tệp trong thư mục
- Step 2 Xoá thư mục
- Step 3 Kết thúc

#### *2. Phương pháp thực hiện mục đích xoá mọi tệp*

- Step 1 Chọn lệnh ERASE từ tệp lệnh
- Step 2 Nghĩ tên thư mục
- Step 3 Nhập tên thư mục: \*.\*
- Step 4 Nhập lệnh và thực hiện lệnh
- Step 5 Kết thúc

#### *3. Phương pháp thực hiện mục đích chuyển thư mục*

- Step 1 Chọn lệnh RD từ tệp lệnh
- Step 2 Nghĩ tên thư mục và nhập
- Step 3 Nhập lệnh và thực hiện lệnh
- Step 4 Kết thúc

### **A.3 Các phương pháp quản lý tệp - 3**

#### *1. Phương pháp thực hiện mục đích chuyển 1 thư mục*

- Step 1 Thực hiện sao thư mục
- Step 2 Thực hiện xoá thư mục cũ
- Step 3 Kết thúc

#### *2. Phương pháp thực hiện mục đích sao 1 thư mục*

- Step 1 Tạo thư mục mới
- Step 2 Sao mọi tệp sang thư mục mới
- Step 3 Kết thúc

**3. Phương pháp thực hiện mục đích sao mọi tệp**

- Step 1 Chọn lệnh COPY từ tệp lệnh
- Step 2 Nghĩ tên thư mục nguồn
- Step 3 Nhập \*.\* trong vùng nguồn
- Step 4 Nghĩ tên thư mục đích
- Step 5 Nhập \*.\* trong vùng đích
- Step 6 Thực hiện lệnh
- Step 7 Kết thúc

**B. Với máy Macintosh**

**B.1. Các phương pháp quản lý tệp cụ thể**

**1. Phương pháp thực hiện mục đích xoá 1 tệp**

- Step 1 Gấp tệp bỏ vào thùng rác
- Step 2 Kết thúc

**2. Phương pháp thực hiện mục đích chuyển 1 tệp**

- Step 1 Gấp tệp bỏ vào nơi đến
- Step 2 Kết thúc

**3. Phương pháp thực hiện mục đích xoá 1 thư mục**

- Step 1 Gấp thư mục bỏ vào thùng rác
- Step 2 Kết thúc

**4. Phương pháp thực hiện mục đích chuyển 1 thư mục**

- Step 1 Gấp thư mục bỏ vào nơi đến
- Step 2 Kết thúc

**B.2. Các phương pháp quản lý tệp khái quát**

**1. Phương pháp thực hiện xoá 1 đối tượng**

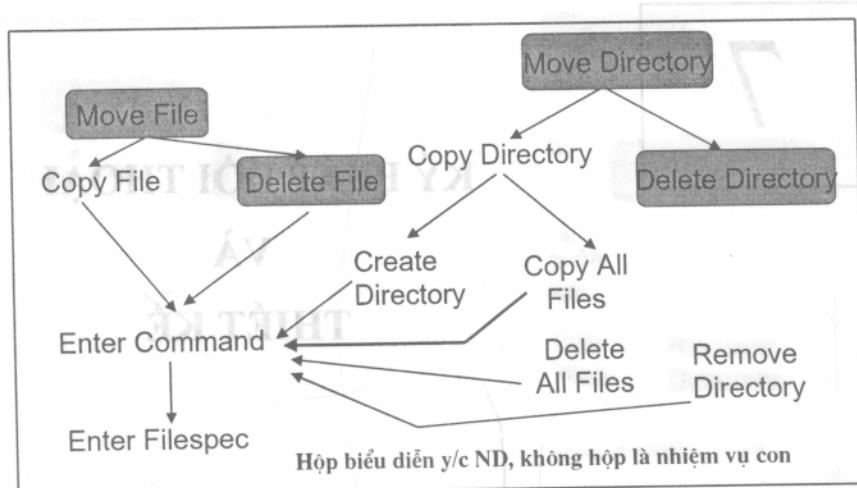
- Step 1 Gấp đối tượng bỏ vào thùng rác
- Step 2 Kết thúc

**2. Phương pháp thực hiện chuyển 1 đối tượng**

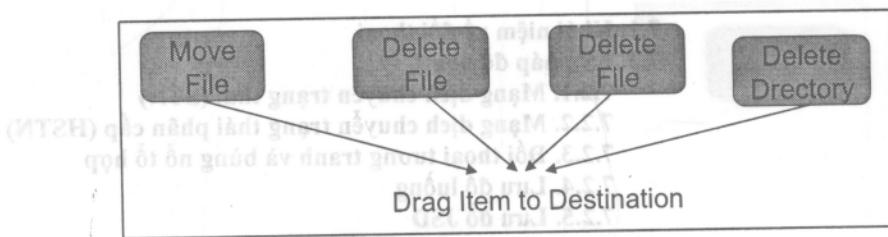
- Step 1 Gấp đối tượng bỏ vào nơi đến
- Step 2 Kết thúc

Hai lưu đồ dưới đây minh họa toàn bộ cấu trúc mục đích của yêu cầu quản lý tệp và tập các nhiệm vụ (hành động) phải thực hiện để đạt mục đích cũng như mối liên hệ giữa chúng.

## Chương 6: Đặc tả yêu cầu và phân tích nhiệm vụ



Cấu trúc mục đích của PC- MSDOS



Cấu trúc mục đích của Macintosh

**7**

**KÝ PHÁP ĐỐI THOẠI  
VÀ  
THIẾT KẾ**

**Nội dung**

- 7.1. Khái niệm về đối thoại**
- 7.2. Ký pháp đồ họa**
  - 7.2.1. Mạng dịch chuyển trạng thái (STN)**
  - 7.2.2. Mạng dịch chuyển trạng thái phân cấp (HSTN)**
  - 7.2.3. Đối thoại tương tranh và bùng nổ tổ hợp**
  - 7.2.4. Lưu đồ luồng**
  - 7.2.5. Lưu đồ JSD**
- 7.3. Ký pháp văn bản**
  - 7.3.1. Văn phạm**
  - 7.3.2. Luật sản xuất**
  - 7.3.3. CPS đại số và sự kiện**
- 7.4. Ngữ nghĩa đối thoại và thí dụ**

Đối thoại là mức độ ngữ nghĩa của HCI và khá giống với một vở diễn, trừ người dùng và đôi khi cả máy tính và có khá nhiều lựa chọn. Ký pháp dùng để mô tả đối thoại có thể là:

- i) Lưu đồ: dễ dàng linh hoạt
- ii) Văn bản: Dễ hơn cho phân tích hình thức

Đối thoại liên kết với:

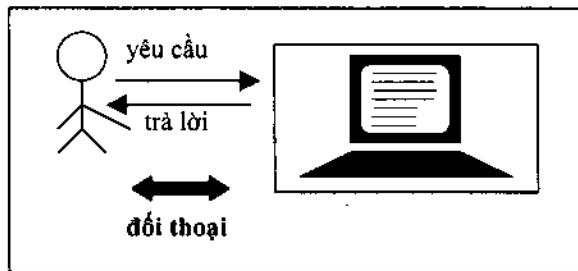
- i) Ngữ nghĩa của hệ thống mà nó thực hiện
- ii) Biểu diễn của hệ thống: dáng vẻ như thế nào?

Mô tả hình thức có thể dùng phân tích cho:

- i) các hành động không nhất quán
- ii) khó khăn của các hành động dự trữ
- iii) các mục cὸn thiếu
- iv) các lỗi tiềm ẩn

### 7.1. Đối thoại là gì?

Đối thoại khác với độc thoại, là cuộc trao đổi giữa hai đối tác. Trong ngữ cảnh HCI, hai đối tác đó là người dùng con người và máy tính. Việc đối thoại giữa người dùng và MT có thể minh họa qua hình 7.1 dưới đây:



Hình 7.1. Đối thoại người dùng - máy tính.

Thường chúng ta hay xem xét ba mức độ của một ngôn ngữ máy tính:

- 1) *Mức độ từ vựng*: là mức độ thấp nhất. Đó là hình dạng, biểu tượng, phím nhấn.
- 2) *Mức độ cú pháp*: Thứ tự và cấu trúc của đầu vào, đầu ra (trong ngôn ngữ tự nhiên đó là ngữ pháp xây dựng câu).
- 3) *Mức độ ngữ nghĩa*: ý nghĩa của trao đổi theo thuật ngữ hiệu quả trên cấu trúc dữ liệu bên trong của máy tính và hoặc với thế giới bên ngoài.

Trong giao tiếp ND (User Interface), thuật ngữ đối thoại thường đồng nghĩa với mức độ cú pháp.

#### 7.1.1. Đối thoại người dùng có cấu trúc

Ngược với đàm thoại người dùng, đối thoại với máy tính thường có cấu trúc và bị ràng buộc. Tuy nhiên cũng có những đàm thoại có cấu trúc và thường diễn ra khi có nhiều bên (con người) tham gia.

- Các thành viên có thể trả lời những câu đã xác định trước. Tuy nhiên cũng có thể phụ thuộc các tình huống khác nhau, không lường trước.
- Trong giao tiếp Người - Máy, nhiều khi cũng không xác định hết các tình huống và nhiều khi máy đưa ra các thông báo khó lường; cú pháp liên kết với ngữ nghĩa ứng dụng.

### 7.1.2. Ký pháp biểu diễn đối thoại

Một số các kỹ sư máy tính khá quen thuộc với một số ký pháp, vậy tại sao lại cần đến ký pháp đặc biệt? Hơn nữa chúng ta đã có các ngôn ngữ lập trình thì tại sao không dùng chúng?

Vấn đề là ở chỗ các ngôn ngữ lập trình với các cấu trúc không đủ để mô tả. Chúng ta muốn tách riêng phân tích và tách các phần tử giao tiếp của chương trình khỏi ngữ cảnh tính toán của nó.

## 7.2. Ký pháp đồ họa

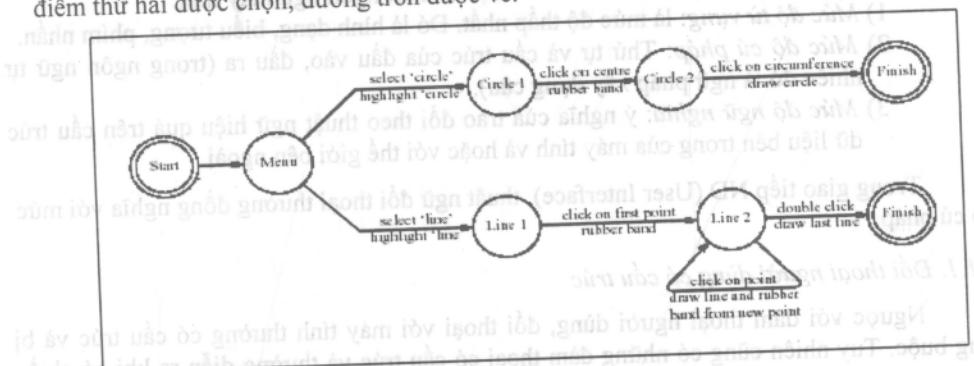
Ký pháp đồ họa hay các sơ đồ thường được dùng trong biểu diễn đối thoại do tính trực quan. Tuy nhiên, nó không phải luôn tốt nhất là đối với các đối thoại phức tạp. Dưới đây chúng ta sẽ xem xét một số ký pháp tiêu biểu.

### 7.2.1. Mạng dịch chuyển trạng thái - STN

Mạng dịch chuyển trạng thái đã được sử dụng từ rất sớm để mô tả đối thoại (1960). Trong ký pháp này, người ta thường dùng hai đối tượng: vòng tròn và mũi tên để mô tả.

- Vòng tròn: mô tả một trạng thái của hệ thống.
- Mũi tên: mô tả dịch chuyển trạng thái - hành động hay sự kiện. Trên mũi tên có thể có nhãn..

Hãy xem xét thí dụ về một công cụ vẽ đơn giản dựa vào menu (hình 7.2). Menu chỉ có hai lựa chọn: vẽ vòng tròn và vẽ đường. Nếu người dùng chọn vẽ đường tròn, họ có thể chọn thêm hai điểm trên không gian vẽ. Điểm thứ nhất biểu diễn tâm đường tròn, và điểm thứ hai là một điểm bất kỳ nằm trên đường tròn. Sau khi chọn điểm thứ nhất, hệ thống sẽ vẽ một đường nối tâm đường tròn với vị trí hiện thời của con trỏ trên không gian vẽ. Sau khi điểm thứ hai được chọn, đường tròn được vẽ.



Hình 7.2. Mạng STN cho công cụ vẽ dùng menu.

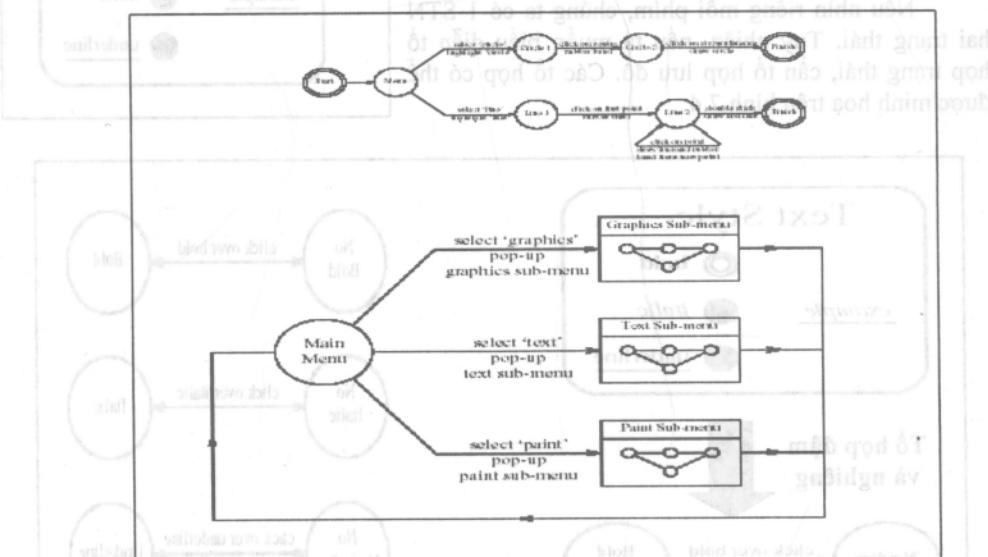
## Chương 7: Ký pháp đối thoại và thiết kế

Hệ thống hoạt động như sau:

- "Menu" là trạng thái trong đó hệ thống đợi người dùng nhấn "circle" hay "line" từ menu.
- "Circle2" là trạng thái sau khi người dùng chọn vòng tròn trung tâm và đợi một điểm trên vòng ngoài.
- "Circle1" là trạng thái hệ thống đợi người dùng chọn vòng tròn trung tâm. Nếu người dùng chọn 1 điểm, hệ thống chuyển sang trạng thái "Circle2".
- Trạng thái "Start", "Finish" là các trạng thái không thật. Nó giúp cho ta dễ theo dõi khi biểu diễn đối thoại phức tạp.

### 7.2.2. Mạng dịch chuyển trạng thái phân cấp - HSTN

Được sử dụng khi đối thoại khá phức tạp. Người ta chia đối thoại thành các đối thoại nhỏ (sub-dialog). Thí dụ như main menu có 3 lựa chọn (3 sub menu) được biểu diễn bởi một menu đồ thị và có menu văn bản (hình 7.3).



Hình 7.3. Mạng STN phân cấp cho menu vẽ phức tạp.

Cách biểu diễn này cũng tương tự như STN. Tuy nhiên, để biểu diễn một trạng thái phức tạp, người ta sử dụng thêm các STN biểu diễn bởi một hình chữ nhật. Hoạt động của hệ thống được hiểu như sau:

Chúng ta bắt đầu từ trạng thái mainmenu và tiếp theo một trong các trạng thái có thể. Quá trình tiếp theo như sơ đồ hình 7.2. Tuy nhiên, trong mỗi khối lại là các sub menu tương đương một trạng thái phức hợp.

Việc sử dụng HSTN không làm thay đổi sức mạnh của các ký pháp cơ sở, song nó làm cho việc biểu diễn các hệ thống lớn phức tạp trở nên đơn giản hơn nhờ sự phân cấp. Sử

## Chương 7: Ký pháp đối thoại và thiết kế

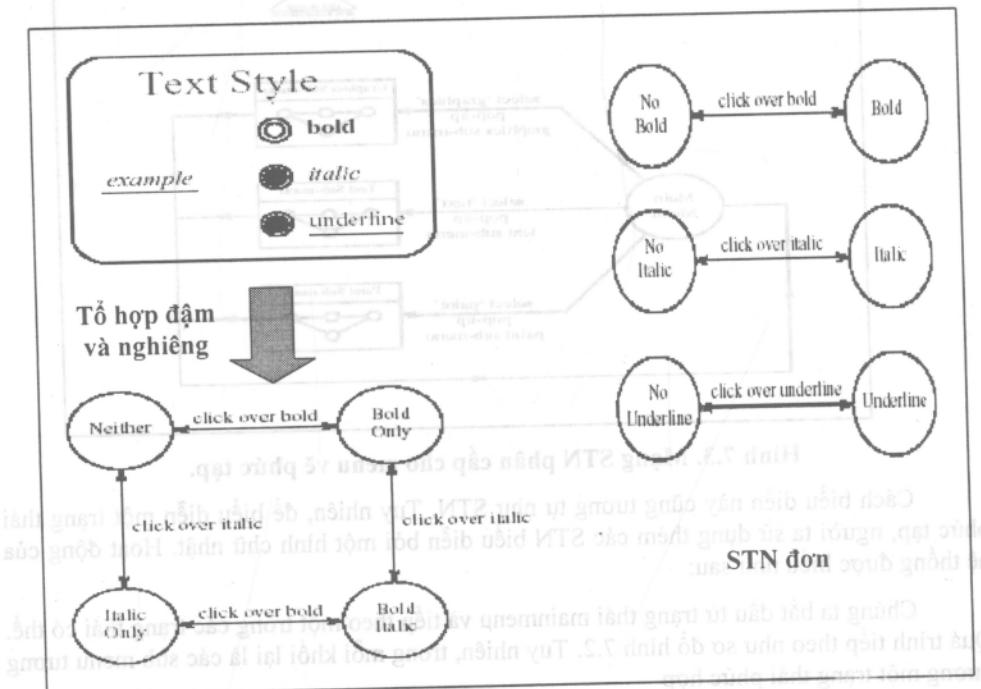
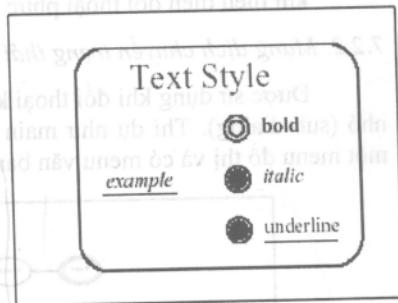
dụng STN có thể là một xuất phát tốt để tạo ra mẫu thử. Trong trường hợp đơn giản nhất có thể dùng giấy: vẽ bằng tay hay in từ máy tính. Hơn nữa, ta có thể duyệt toàn bộ kịch bản với người dùng hay khách hàng và giải thích nhờ STN. Trong một số trường hợp khác có thể dùng máy tính với Hypercard hay Macromedia Director để vẽ mỗi trạng thái.

### 7.2.3. Đối thoại tương tranh và bùng nổ tổ hợp

STN có thể rất tốt khi biểu diễn đối thoại tuần tự, chọn hay lặp. Tuy nhiên, nó sẽ dàn kém đi tác dụng nếu phải biểu diễn các phần tương tranh. Để hiểu rõ hơn, xét thí dụ một hội thoại đơn giản kiểu văn bản trong một hệ STVB.

Hộp lựa chọn chứa 3 phím Toggle. Mỗi phím cho 1 tình huống: đậm, nghiêng và gạch chân. Một đoạn văn bản có thể là nghiêng, đậm hay gạch dưới và cũng có thể là bất kỳ tổ hợp nào của 3 thuộc tính trên.

Nếu nhìn riêng mỗi phím, chúng ta có 1 STN hai trạng thái. Tuy nhiên, nếu ta muốn biểu diễn tổ hợp trạng thái, cần tổ hợp lưu đồ. Các tổ hợp có thể được minh họa trên hình 7.4.

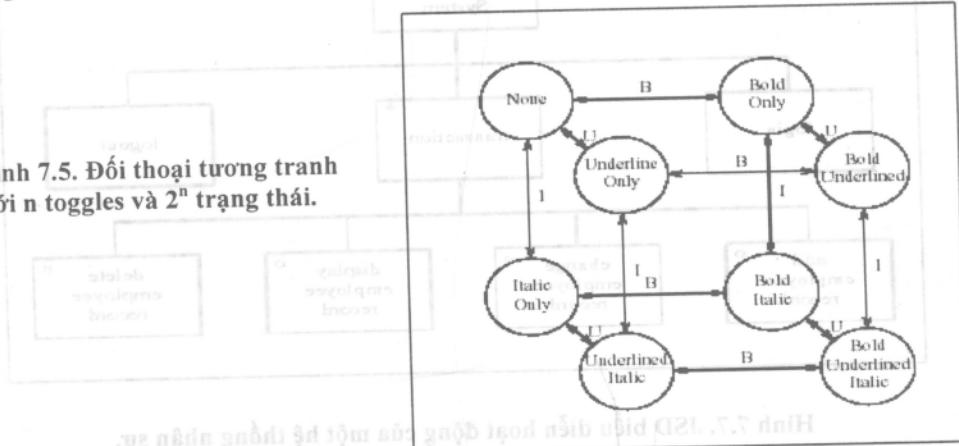


Hình 7.4. Tương tranh và tổ hợp.

## Chương 7: Ký pháp đối thoại và thiết kế

Với một số đối thoại phức tạp, người ta phải dùng dạng khối không gian. Với  $n$  toggles ta có  $2^n$  trạng thái (hình 7.5).

Hình 7.5. Đối thoại tương tranh với  $n$  toggles và  $2^n$  trạng thái.



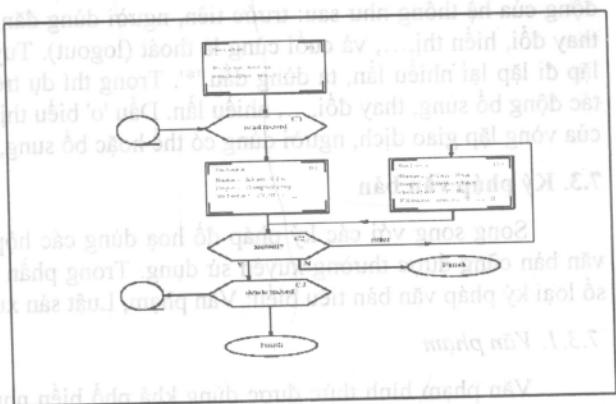
### 7.2.4. Lưu đồ luồng - Flow Chart

Lưu đồ luồng là một công cụ khá quen thuộc với Người lập trình. Các thành phần của lưu đồ là các hộp hình chữ nhật, hộp đa giác, các mũi tên và một số ký pháp vật lý khác. Chú ý rằng:

**Hộp** là quá trình hay là một quyết định.

- Sự kiện không phải là trạng thái.

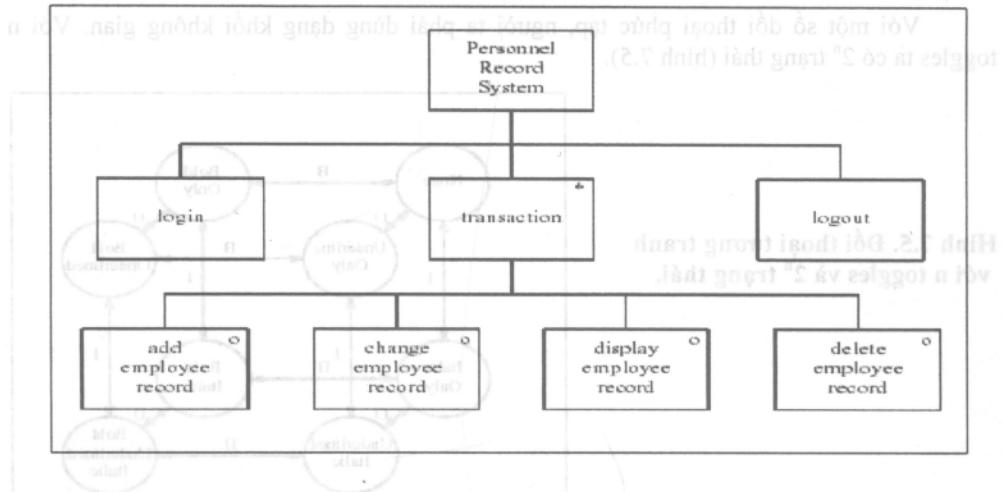
**Sử dụng nhiều loại hộp khác nhau để biểu diễn các hoạt động khác nhau và nó quan hệ** kiều hướng lập trình hơn là hướng người dùng (hình 7.6).



### 7.2.5. Lưu đồ Jackson - JSD (Jackson Structured Design)

Ra đời sau lưu đồ luồng, song được dùng trong nhiều khía cạnh của TA. Lưu đồ JSD thích hợp với cấu trúc đối thoại hình cây. Tuy kém diễn tả song sáng sủa hơn. Nó gần giống như HTA, song chính xác hơn. Hình 7.7 diễn tả một hệ thống nhân sự, hệ thống cho phép cập nhật thông tin về nhân sự: bổ sung, hiện, xoá,...

## Chương 7: Ký pháp đối thoại và thiết kế



Hình 7.7. JSD biểu diễn hoạt động của một hệ thống nhân sự.

Ta nhận thấy trong một vài hộp có chứa dấu '\*' hay dấu 'o' ở góc phải trên. Hoạt động của hệ thống như sau: trước tiên, người dùng đăng nhập (login), tiếp theo là bổ sung, thay đổi, hiển thị,..., và cuối cùng là thoát (logout). Tuy nhiên, để biểu thị hành động được lặp đi lặp lại nhiều lần, ta dùng dấu '\*'. Trong thí dụ trên, người dùng có thể thực hiện các tác động bổ sung, thay đổi,..., nhiều lần. Dấu 'o' biểu thị sự lựa chọn: trong một lần tác động của vòng lặp giao dịch, người dùng có thể hoặc bổ sung, hoặc thay đổi,...

### 7.3. Ký pháp văn bản

Song song với các ký pháp đồ họa dùng các hộp, các đường, mũi tên, các ký pháp văn bản cũng được thường xuyên sử dụng. Trong phần dưới đây, chúng ta sẽ xem xét một số loại ký pháp văn bản tiêu biểu: Văn phạm, Luật sản xuất và CPS đại số.

#### 7.3.1. Văn phạm

Văn phạm hình thức được dùng khá phổ biến như một ký pháp văn phạm, trong đó BNF (Backus-Naur-Form) là một dạng hay dùng như trong mô tả ngôn ngữ lập trình và trong mô tả giao tiếp. Ký pháp chung của BNF có dạng:

<đối tượng> ::= <giải nghĩa 1>/<giải nghĩa 2>/<giải nghĩa 3>/...

Thí dụ, để định nghĩa một biểu thức chính quy, ký pháp BNF cho định nghĩa này có dạng:

expr ::= empty | atom expr | '(' expr ')' expr

Định nghĩa này được hiểu rằng: **biểu thức** có thể là **rỗng** hay một **biểu thức nguyên tố** hay là một **biểu thức**.

Ưu điểm của ký pháp này là đa dạng hơn so với **biểu thức chính quy** hay STN. Tuy nhiên, nó không có khả năng biểu diễn tranh.

## **Chương 7: Ký pháp đối thoại và thiết kế**

### **7.3.2. Luật sản xuất**

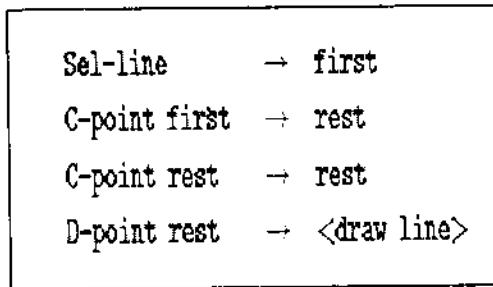
Chúng ta đã xem xét luật sản xuất trong chương 6 như là một phần của mô hình CCT hay trong mô tả thành phần đối thoại của UIMS (chương 8). Luật sản xuất thường có dạng:

**If cond then action, và có thể viết cond -> action hay cond: action**

Luật hoạt động theo nguyên tắc: Các luật được kích hoạt và hệ thống sánh phần cond của luật với các sự kiện khởi tạo bởi người dùng. Khi cond là thỏa, luật gọi là “fire” và hành động được thực hiện. Hành động có thể là dạng đáp ứng của người dùng hay sự thay đổi trạng thái của hệ thống. Hệ thống luật sản xuất có thể là hướng sự kiện hay hướng trạng thái và cũng có thể là cả hai.

Chú ý: Thứ tự xuất hiện của luật là không quan trọng.

Để minh họa, chúng ta lấy lại thí dụ vẽ đường được trình bày trong phần trên (7.2.1). Luật sản xuất cho trường hợp này được minh họa trong hình 7.8.



**Hình 7.8. Luật sản xuất biểu diễn menu đồ họa vẽ.**

Trong thí dụ này, phần điều kiện và phần hành động của luật được thể hiện dưới dạng đơn giản: dạng sự kiện. Người ta thường dùng ba loại sự kiện sau:

- **Sự kiện người dùng:** Thường bắt đầu với chữ hoa. Trong thí dụ trên đó là các phần đầu của luật. Thí dụ sự kiện Sel-line là do người dùng nhấn phím lựa chọn trên menu (bằng bàn phím hay chuột).
- **Sự kiện trong:** thường bắt đầu với chữ thường. Nó được dùng trong đối thoại để lưu lại vết của trạng thái đối thoại. Trong thí dụ trên rest line là trạng thái sau khi điểm đầu tiên được chọn
- **Đáp ứng của hệ thống:** thường bao trong cặp "<>". Đó là hiệu ứng nhìn thấy hoặc nghe thấy của hệ thống.

Một luật là “fire” nếu mọi sự kiện trong phần điều kiện của nó hiện diện trong bộ nhớ và mọi tương tác với người dùng là được thực hiện ngay lập tức bởi sự kiện bộ nhớ này: sự kiện người dùng như nhấn chuột được bổ sung vào bộ nhớ và đáp ứng của hệ thống là đường được vẽ (<draw line>).

## Chương 7: Ký pháp đối thoại và thiết kế

Khi một luật là fire, 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. Như trong thí dụ trên khi người dùng chọn Sel-line từ menu, bộ nhớ của hệ thống sẽ chứa Sel-line. Điều này có nghĩa là luật thứ nhất có thể là fire, Sel-line sau đó có thể bị loại khỏi bộ nhớ và thay thế vào đó là first,...

Luật sản xuất hướng trạng thái do Olsen đề xuất lại có một hành vi rất khác. Tuy rằng hệ thống bộ nhớ của nó cũng chứa một tập các giá trị tên song tập tên này không bị loại khỏi bộ nhớ (ngầm định) khi một luật là fire. Nó chỉ bị loại bỏ một cách tường minh bởi phần hành động của luật.

Quay lại thí dụ mô tả trong phần 7.2.1. Chúng ta có 5 thuộc tính với thí dụ này:

- **Mouse**: mouse off, select line, click point, double click
- **Line state**: menu, start-line, rest-line
- **Rubber-band**: rubber-band off, rubber-band on
- **Menu**: highlight off, highlight-line, highlight-circle
- **Draw**: draw nothing, draw line.

Thuộc tính thứ nhất "mouse" được thiết lập tự động khi người dùng thực hiện một hành động. Thuộc tính thứ hai được dùng bởi đối thoại để lưu lại vết, và ba thuộc tính còn lại điều khiển đáp ứng của hệ thống.

Luật sản xuất cho thí dụ này cũng tương tự như trong trường hợp hướng sự kiện:

- select-line -> mouse off, start-line, highlight-line
- click point start-line -> mouse off, rest-line, rubber-band on
- click point rest-line -> mouse off, draw line
- double click rest-line -> mouse off, menu, draw line, rubber-band off

Nhìn chung luật sản xuất thích hợp cho mô tả tương tranh: Luật dựa trên sự kiện. Nó không thích hợp cho tuần tự hay trạng thái.

### 7.3.3. CPS đại số và sự kiện

Ký pháp STN rất phù hợp với việc điều khiển tương tranh, trong khi đó luật sản xuất lại ngược lại. Việc xử lý đồng thời (tương tranh, tuần tự) đặt ra trong nhiều bài toán (truyền thông, điều khiển tương tranh). Quá trình đại số (Process Algebras) là một lớp ký pháp hình thức được phát triển cho các quá trình tương tự. CSP là một lớp con được phát triển cho đặc tả đối thoại và đặc tả cho cả tuần tự và tương tranh.

Thí dụ mô tả cho menu đồ họa trên được mô tả như sau:

Draw-menu = (select - circle? -> Do-circle  
                 [] select-line? -> Do-line)

Do-circle = click? -> set centre -> click ? Draw- circle -> Skip

Do-line = Start-line ; Rest-line

Start-line = click? -> first-point -> Skip

Rest-line = (click? ->next-point -> Rest-line [] double-click? ->last-point ->Skip

## *Chương 7: Ký pháp đối thoại và thiết kế*

với quy ước:

- Các sự kiện được đánh dấu hỏi là các hành động của người dùng.
- dấu “=” có nghĩa là định nghĩa
- ->: sự kiện tuần tự
- “;”: quá trình tuần tự
- []: sự lựa chọn. Tên biểu diễn sự kiện.
- Do-circle là hoàn toàn tuần tự. Khi hệ thống thực hiện Do-circle, trước tiên cần người dùng nhấn phím chuột, tiếp sau là một sự kiện trong “set centre” để xác định vị trí con trỏ. Tiếp theo nhận một lần nhấn chuột rồi về và kết thúc bởi Skip.
- Do-line cũng là tuần tự. Dấu “;” để chỉ quá trình tuần tự, cái xảy ra giữa 2 quá trình. Dấu “->” chỉ dùng sau một sự kiện.
- []: chỉ ra sự lựa chọn như trong dòng một: người dùng có thể chọn circle hay chọn line.
- Quá trình tương tranh được dùng như một cách thức để tổ chức cấu trúc trong của giao tiếp. Thí dụ khi lựa chọn, người dùng có thể dùng chuột hay dùng phím nóng => mỗi lựa chọn một quá trình.
- Quá trình chuột đơn giản là đợi ND chọn một mục trên menu và tiếp sau là một sự kiện trong phụ thuộc vào sự lựa chọn.
- Quá trình bàn phím điều khiển bởi phím Alt và tiếp sau cũng là một sự kiện trong.

### Xét thí dụ minh họa bằng tính cộng

Trong thí dụ này ta dùng ký pháp CSP để minh họa một bảng tính chỉ giới hạn bởi phép cộng. Ba tính huống có thể:

- 1) người dùng nhập zero: khởi tạo tổng bằng 0 để bắt đầu (Reset)
- 2) người dùng nhập 1 số: sẽ cộng thêm vào tổng
- 3) người dùng nhấn “quit”: Kết thúc.

Đối thoại được miêu tả:

```
Cộng = add-prompt -> (quit? -> skip
    [] zero? -> show-sum -> Cộng
    [] number? -> show-sum -> Cộng
```

Với ngữ nghĩa miêu tả như sau:

```
event add prompt = prompt: true      out: "next number"
event quit? =
    ues: input  when input = "quit"
event zero? =
    ues: input when input = "zero" set sum = 0 out: "reset"
event number? =
    ues: input, sum when is number (input)
```

```

    set sum = sum + input
event show-sum =
    uses: sum  out: sum

```

Chúng ta xét thêm một thí dụ: thí dụ về **cơ sở dữ liệu**:

Người dùng có thể sử dụng các hành động: "quit", "set key value", hay "Find key".

Đối thoại được miêu tả:

```

DB = db-prompt -> (quit? -> skip
[] set? -> get-key; get-val
[!get? -> get-key; print-val)
get-key = set-key-prompt -> get-key?
get-val = set-val-prompt -> get-val?

```

Ngữ nghĩa đối thoại tương tự như phần cộng ở trên. Ta chỉ minh họa cho những điều khác như "get":

```

event get-key? =
    uses: input  set: key = input event get-val? =
        uses: input, dbase, key
        set: dbase = db-update(dbase, input, key)
event print-val? =
    uses: dbase, key
    out: db-lookup(dbase, key)

```

#### 7.4. Ngữ nghĩa đối thoại và thí dụ

Nếu mục đích của mô tả đối thoại chỉ đơn giản là thông báo giữa các nhà thiết kế hay như là một "công cụ để suy nghĩ" thì chỉ cần chú giải cho đối thoại hình thức với chủ ý của hành động hoặc để cho đọc giả tự suy luận về ngữ nghĩa. Tuy nhiên, nếu mô tả đối thoại dùng cho một đặc tả hình thức cho một hợp đồng hoặc để thực hiện một mẫu thử thì cần phải miêu tả một cách hình thức đối thoại này bằng nhiều cách khác nhau. Ký pháp đối thoại mà chúng ta đã xem, ít hay nhiều phải mô tả rõ ràng cấu trúc của đối thoại.

Có hai khía cạnh của ngữ nghĩa đối thoại chúng ta cần xem xét. Thứ nhất là đối với ứng dụng và thứ hai đối với việc trình diễn. Phần ngữ nghĩa của đối thoại nên dùng như một mối liên hệ giữa ứng dụng và trình diễn như nó có thể càng ít càng tốt. Như trong thí dụ trên (phần 7.2.1), đường được vẽ bởi công cụ vẽ được thực hiện ở các điểm khi người dùng nhấn chuột. Hoặc trong luật sản xuất sau:

```
click_at(x,y) -> dot_at(c,y), call another_point(x,y)
```

có nghĩa là khi người dùng nhấn chuột trong vùng vẽ, hệ thống vẽ một điểm tại vị trí chuột và kéo theo lời gọi vẽ điểm tiếp another\_point.

Một khía cạnh khác cũng cần xem xét đó là ký pháp đối thoại thường gắn với ngôn ngữ lập trình. Thí dụ "input tool" dưới đây là một ký pháp dựa vào biểu thức chính quy sử dụng ngôn ngữ C để thể hiện ngữ nghĩa đối thoại.. Một đoạn code C mô tả việc đọc một số được viết như sau:

```

tool number
{
    char buf[80];
    int index;
    int positive;
    input { ( digit * + sign; digit; digit*); return}
    tool digit
    {
        input {key: | key_c >='0' && key_c <='9'}
        if (index < 79)
            { buf[index] = key_c;
              inde = inde +1;
              echo(key_c); }
    }
    tool sign
    .....
    tool return
    { input {key: | key_c=='\n' |}
    .....
}
.....
}

```

Input tool sử dụng cú pháp của biểu thức chính quy và hai phép toán bổ sung. Trình tự được thể hiện bởi ';' và việc chọn thể hiện bởi '+'. Biểu thức 'key: | cond |' là hậu điều kiện của biểu thức: biểu thức chỉ sánh được nếu điều kiện là thoả.

Việc đặc tả biểu thức 'input tool' có nghĩa như sau: Từ khoá *tool* để chỉ ra một công cụ mới, có ý nghĩa như ký tự không kết thúc trong ký pháp BNF và biểu thức chính quy biểu diễn nó được bao bọc trong câu lệnh *input*. Tool được sắp theo thứ tự phân cấp, có nghĩa là *digit, sign, return* là của riêng *number*. Lệnh *Echo* chỉ đơn giản là thông báo quay ngược lại người dùng. Cuối cùng, *key* là công cụ (primitive) sánh một ký tự đơn đọc từ bàn phím, ký tự hiện thời được lưu trữ trong biến toàn cục *key\_c*. Ở đây cần chú ý là các sub tool như *digit* thông báo kết quả cho main tool thông qua biến dùng chung *buf*.

### Kết luận

Chúng ta thấy rằng nhiều ký pháp khác nhau song lại có những điểm tương đồng chủ yếu. Thí dụ như lưu đồ JSD và biểu thức chính quy có thể cùng mô tả chính xác một đối thoại. Tuy nhiên, sức mạnh biểu diễn của các công cụ khác nhau cũng rất khác nhau. Thi dụ như ký pháp BNF khá mạnh khi biểu diễn các đối thoại lồng nhau và đệ quy hơn là ký pháp STN.

Điều khác nhau quan trọng trong sức mạnh biểu diễn là ở chỗ ký pháp đó có thể hay không có thể biểu diễn đối thoại tương tranh. Một số thì nghiêng về tuần tự, một số khác lại nghiêng về tương tranh và rất ít ký pháp làm được cả hai.

Khi chúng ta đã có một miêu tả đối thoại, cần xem xét tính liên thông giữa miêu tả đối thoại với việc miêu tả ngữ nghĩa của hệ thống.

# 8

## HỖ TRỢ CÀI ĐẶT

### Nội dung

- 8.1. Tổng quan
- 8.2. Các thành phần của hệ thống Window
- 8.3. Lập trình ứng dụng
- 8.4. Sử dụng công cụ (Toolkits)
- 8.5. Hệ quản trị giao tiếp người dùng (UIMS)

## Chương 8: Hỗ trợ cài đặt

### 8.1. Tổng quan

Cài đặt phụ thuộc vào một hệ thống cụ thể, do vậy chúng ta không cần phải xem xét một cách chi tiết các thiết bị tham gia tương tác. Phần này chỉ mô tả một số công cụ hỗ trợ như các ngôn ngữ lập trình, các Tools, Toolkit hay hệ quản trị giao tiếp người dùng UIMS. Các chức năng giao tiếp của các hệ thống thường được hỗ trợ bởi các module vào ra hay các ngôn ngữ lập trình, các hỗ trợ của hệ thống windows.

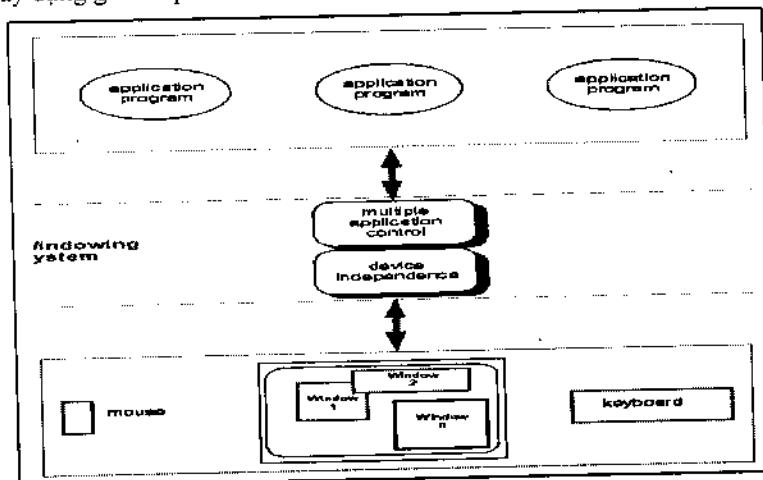
Chúng ta cần chú ý các vấn đề:

- i) Tính độc lập với ứng dụng
- ii) Tính độc lập với các thiết bị vào ra
- iii) Tính độc lập với người dùng.

Công cụ lập trình cung cấp các dịch vụ cho người lập trình. Hệ thống window là lõi cho các luồng xử lý hệ thống - người dùng đồng thời hay riêng biệt. Lập trình trong window giúp cho người dùng không bận tâm tới các hàm vào ra nguyên thuỷ (primitives) của máy mà ứng dụng cần tới. Các công cụ giao tiếp tạo cho chương trình gần với mức cảm nhận của người dùng. Các UIMS giúp cho việc điều khiển các mối quan hệ giữa biểu diễn và chức năng của các đối tượng.

### 8.2. Các thành phần của hệ thống Window

Trong các chương trước (chương 3) chúng ta đã đề cập đến giao tiếp WIMP một loại giao tiếp chủ yếu, chuẩn được dùng trong HCI. Tuy nhiên, trong chương đó chúng ta mới dừng ở việc xem xét cách thức mà giao tiếp này tăng cường cho tương tác với người dùng. Trong phần này chúng ta sẽ xem xét một cách chi tiết hơn cách hệ thống windows được sử dụng để xây dựng giao tiếp WIMP.



Hình 8.1. Vai trò của hệ thống window.

Như chỉ ra trên hình 8.1, hệ thống window như một cầu nối giữa các chương trình ứng dụng và các thiết bị vật lý đầu cuối mà người dùng dùng để tương tác.

Đặc trưng quan trọng thứ nhất của hệ thống window là khả năng cung cấp tính độc lập thiết bị cho người lập trình. Mọi trạm làm việc tiêu biểu đều có một số thiết bị chung như thiết bị hiển thị, bàn phím, thiết bị định vị trỏ như chuột. Mọi thiết bị này có thể khác nhau song đều được dùng trong các hệ thống giao tiếp và nói chung chúng rất nhau trong cách truyền dữ liệu dĩ cũng như cách ra lệnh cho chúng. Điều bắt buộc là khi lập trình cho ứng dụng phải tính đến việc ứng dụng phải hoạt động được trên nhiều loại thiết bị phần cứng khác nhau, hay nói cách khác là tính độc lập với thiết bị.

Để làm được điều này, các nhà lập trình hướng các lệnh đến một thiết bị đầu cuối trừu tượng (abstract terminal) có thể hiểu được một ngôn ngữ khai quát và có thể được dịch sang một ngôn ngữ của một số thiết bị chuyên dụng nào đó. Điều này làm cho việc lập trình dễ hơn và cũng tăng tính khả chuyển của ứng dụng: ta chỉ cần viết cho một thiết bị cụ thể nào đó và sau đó một chương trình ứng dụng có thể truy cập đến.

Một hệ thống window thường có một ngôn ngữ khai quát cố định cho thiết bị đầu cuối trừu tượng gọi là mô hình ảnh. Mô hình này sử dụng các mô hình vào ra đồ họa như: điểm ảnh, GKS (Graphic Kernel System), giao tiếp phân cấp lập trình cho đồ họa, (PHIGS), PostScript độc lập với thiết bị phần cứng

Đặc trưng thứ hai của hệ thống window là khả năng chia sẻ tài nguyên: với một cấu hình phần cứng, hệ thống window có thể sao thành nhiều bản cho các thiết bị đầu cuối trừu tượng. Mỗi thiết bị này có một trạng thái như một quá trình độc lập và hệ thống window sẽ cộng tác điều khiển quá trình tương tranh này. Sự cộng tác của các nhiệm vụ đồng thời được nhân lên thành các ứng dụng độc lập và có thể được lập trình riêng như chúng được xử lý riêng rẽ.

**Mô hình ảnh** Bao gồm các phần tử:

- Pixel: coi màn hình được biểu diễn bởi một tập các điểm ảnh (pixel); số điểm ảnh biểu diễn bằng tích số điểm ảnh theo hàng với số điểm ảnh theo cột
- GKS: Chuẩn quốc tế, coi màn hình như một tập các đoạn liên thông, mỗi đoạn là một macro các phần tử đồ họa
- PHIGH: cũng là một chuẩn mở rộng của GKS. Màn hình là dãy các đoạn có thể hiệu chỉnh được
- PostScript: một ngôn ngữ lập trình để mô hình hóa màn hình.

Theo Coutaz và Bass, để các phần mềm có vai trò như hệ thống window, ba kiến trúc sau đây có thể sử dụng:

- 1) Mỗi ứng dụng quản lý tất cả các quá trình của mình
  - mọi quá trình quan tâm đến tính đồng bộ.
- 2) Vai trò quản lý thuộc Hệ điều hành
  - ứng dụng gắn chặt vào Hệ điều hành, các ứng dụng không phải quan tâm đến điều đó. Như vậy ứng dụng phải phát triển dựa vào một hệ điều hành cụ thể.
- 3) Vai trò quản lý như một ứng dụng riêng biệt
  - chức năng quản lý tách riêng như một ứng dụng nhằm cung cấp các giao tiếp cho các ứng dụng khác. Như vậy ta có *kiến trúc client-server (khách - chủ)*.

## Chương 8: Hỗ trợ cài đặt

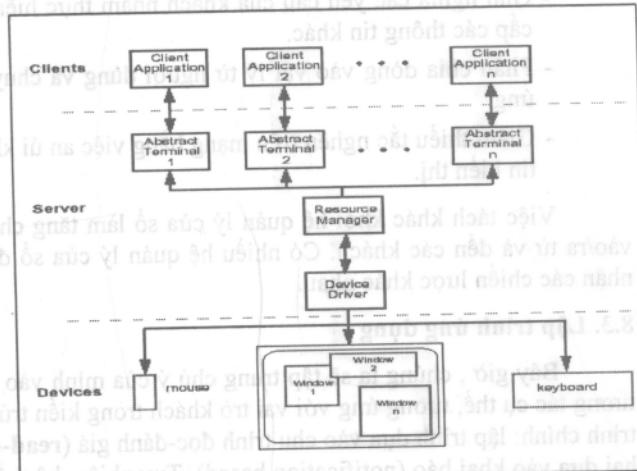
Kiến trúc thứ nhất bộc lộ nhược điểm lớn là mọi quá trình đều phải quan tâm đến tính đồng bộ, trái ngược với quan điểm phân chia tài nguyên. Điều này cũng làm giảm tính khả chuyên của ứng dụng.

Kiến trúc thứ hai bộc lộ điểm yếu là việc phát triển ứng dụng lại phụ thuộc vào một hệ điều hành cụ thể.

Kiến trúc thứ ba khá dễ hơn cả, nó làm tăng tính khả chuyên của ứng dụng khi mà chức năng quản lý được tách khỏi ứng dụng. Kiến trúc Client-Server được minh họa trên hình 8.2.

Kiến trúc client-server là kiến trúc khá phổ biến hiện nay. Nó gồm ba tầng:

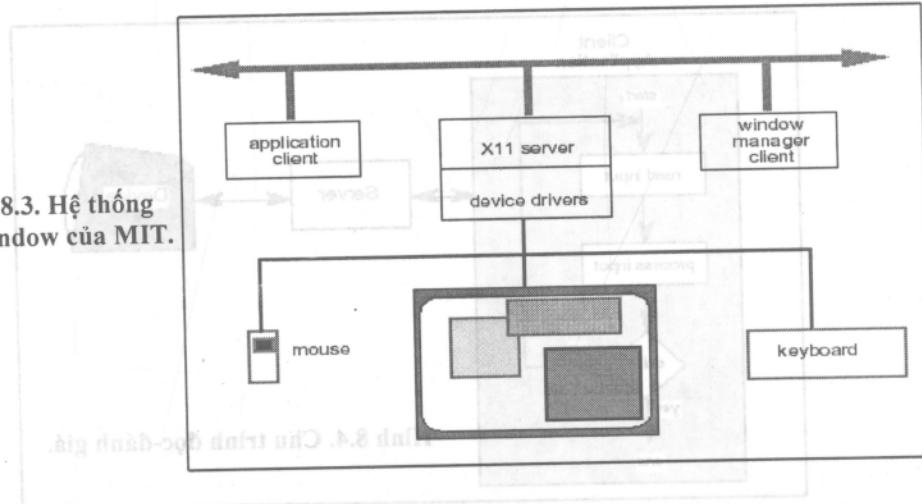
- Tầng thứ nhất là tầng ứng dụng chứa chương trình người dùng.
- Tầng thứ hai: tầng dịch vụ chứa các điều khiển tài nguyên và điều khiển thiết bị
- Tầng thứ ba: tầng vật lý chứa các thiết bị tương tác vào/ra.



Hình 8.2. Kiến trúc client-server.

Một thí dụ kinh điển của hệ thống window theo mô hình client-server là hệ thống X-Window được phát triển tại MIT vào giữa năm 1980. Kiến trúc hệ thống X Window được minh họa trên hình 8.3.

Hình 8.3. Hệ thống X Window của MIT.



X window hay X11 được xây dựng dựa trên mô hình hình ảnh điểm ảnh và giả định có các thiết bị trỏ hỗ trợ. X window dựa vào một giao thức mạng X protocol định nghĩa giao tiếp theo mô hình khách-chủ. Giao thức X có thể cài đặt trên nhiều máy và trên nhiều hệ điều hành khác nhau. Điều này làm tăng tính độc lập thiết bị của X11. Theo kiến trúc này, mỗi khách của X11 liên kết với một thiết bị đầu cuối trùu tượng hay một cửa sổ chính. Máy chủ X thực hiện một tập các nhiệm vụ:

- Cho hay không cho phép truy nhập tới thiết bị hiển thị từ các ứng dụng khách.
- Giải nghĩa các yêu cầu của khách nhằm thực hiện các thao tác màn hình hay cung cấp các thông tin khác.
- Phân chia dòng vào vật lý từ người dùng và chuyển chúng đến khách hàng tương ứng.
- Giảm thiểu tắc nghẽn trên mạng bằng việc an ủi khách do các vết của một số thông tin hiển thị.

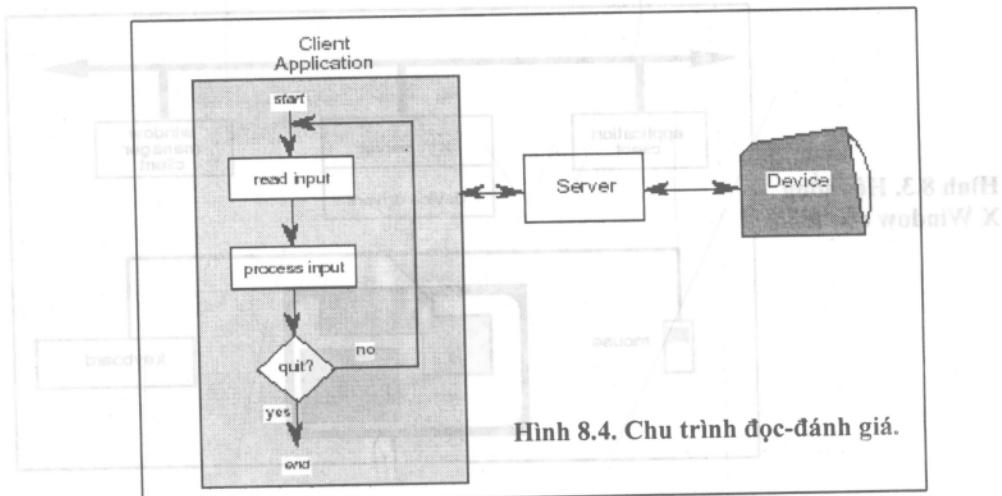
Việc tách khác khỏi hệ quản lý cửa sổ làm tăng chiến lược giải quyết các xung đột vào/ra từ và đến các khách. Có nhiều hệ quản lý cửa sổ được dùng trong X11 và nó chấp nhận các chiến lược khác nhau.

### 8.3. Lập trình ứng dụng

Bây giờ, chúng ta sẽ tập trung chủ ý của mình vào việc lập trình cho một ứng dụng tương tác cụ thể, tương ứng với vai trò khách trong kiến trúc khách-chủ. Có hai mô thức lập trình chính: lập trình dựa vào chu trình đọc-đánh giá (read-evaluation loop) và mô thức thứ hai dựa vào khai báo (notification based). Tuy nhiên, hệ thống window không cần thiết phải xác định xem ứng dụng được cài đặt theo mô thức nào.

#### 8.3.1. Mô thức vòng lặp đánh giá

Mô thức chu trình đọc đánh giá được thể hiện trên hình 8.4. Mô thức lập trình cho Macintosh dựa vào cách này.



Hình 8.4. Chu trình đọc-đánh giá.

## Chương 8: Hỗ trợ cài đặt

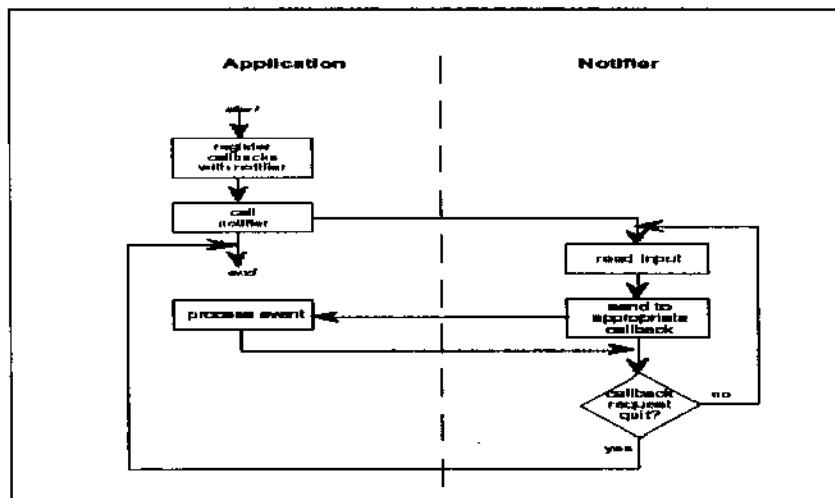
Trong mô thức này, máy chủ X gửi cho ứng dụng khách dòng vào của người dùng dưới dạng các sự kiện có cấu trúc. Điều quan trọng của sự kiện đó là khách hàng mà nó phải hướng tới. Ứng dụng khách được lập trình để đọc một sự kiện chuyên tới cho mình và xác định hành vi của tất cả các ứng dụng đặc biệt mà nó phải trả lại kết quả. Vòng lặp đọc-dánh giá có thể tóm tắt như sau:

```
repeat
    read-event (myevent)
    case myevent.type
        type_1:
            do type_1 processing
        type_2:
            do type_2 processing
        ...
        type_n:
            do type_n processing
    end case
end repeat
```

Theo mô thức này, ứng dụng phải thực hiện giám sát việc xử lý của các sự kiện nó nhận được. Như vậy, người lập trình phải thực hiện việc giám sát trên mỗi sự kiện mà nó sẽ nhận và dẫn đến quá tải.

### 8.3.2. Mô thức thông báo tập trung

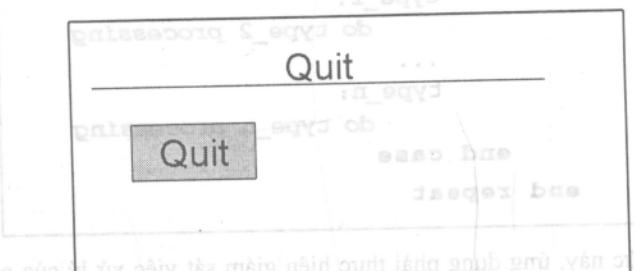
Trong mô thức này, vòng lặp giám sát chính không nằm trong ứng dụng. Bộ Notifier



Hình 8.5. Mô thức Khai báo tập trung (Notification based).

tập trung sẽ nhận các sự kiện từ hệ thống window và lọc chúng rồi gửi cho các trình ứng dụng theo cách khai báo bởi chương trình (hình 8.5). Chương trình ứng dụng thông báo cho notifier biết các sự kiện nó quan tâm. Với mỗi sự kiện, nó khai báo một thủ tục riêng của mình gọi là **Callback** trước khi trả điều khiển lại cho notifier. Khi notifier nhận được một sự kiện từ hệ thống window, nó cần xem xét liệu sự kiện này đã được xác định bởi ứng dụng không. Nếu có, nó chuyển sự kiện và điều khiển cho thủ tục Callback đã đăng ký cho sự kiện này. Sau khi xử lý xong, thủ tục Callback trả lại điều khiển cho notifier hoặc báo cho biết mình sẽ tiếp tục nhận sự kiện hay kết thúc yêu cầu.

Thí dụ, một chương trình tạo một cửa sổ hay một frame trên đó có một phím lệnh “QUIT”. Khi phím này được nhấn bởi thiết bị trò như chuột chương trình sẽ kết thúc và cửa sổ cũng được đóng lại (hình 8.6).



Hình 8.6. Màn hình giao tiếp tạo bởi chương trình.

Màn hình do chương trình tạo ra gồm 3 đối tượng: 1 cửa sổ, thanh tiêu đề và phím lệnh *Quit*. Chương trình ứng dụng sẽ thông báo cho Notifier của thủ tục CallBack được kích hoạt khi phím lệnh được chọn. Chương trình ứng dụng sẽ khởi tạo Notifier bằng việc gọi thủ tục này. Khi Notifier nhận được sự kiện chọn từ phím lệnh, điều khiển sẽ chuyển cho thủ tục *Quit* để đóng cửa sổ và yêu cầu kết thúc.

Với mô thức này, luồng điều khiển tập trung tại notifier và giải phóng cho ứng dụng khỏi công việc tẻ nhạt là phải xử lý mỗi lần khi sự kiện chuyển qua. đương nhiên là sự độc lập này cũng phải trả giá.

Hãy xem thí dụ sau: giả thiết là một chương trình ứng dụng muốn tạo một hộp thông báo khi có sự cố và muốn nhận được thông tin của người dùng trước khi xử lý. Như vậy, hộp hội thoại định trước sẽ tách riêng khỏi chuỗi hành động của người dùng trừ một trường hợp: Sự lựa chọn của người dùng bên trong một vùng nào đó trên màn hình. Để làm điều này, vòng lặp đọc-dánh giá là hầu như tức thì. Giả sử điều kiện lỗi xảy ra trong quá trình xử lý *type2* (xem sơ đồ vòng lặp đọc - đánh giá). Khi điều kiện lỗi được phát hiện và

trình xử lý *type2* (xem sơ đồ vòng lặp đọc - đánh giá). Khi điều kiện lỗi được phát hiện và

nhận dạng, ứng dụng sau đó bắt đầu một chu trình đọc-dánh giá khác chứa trong nhánh của lệnh *case*. Trong vòng lặp này, mọi sự kiện không liên quan có thể nhận được và bị loại bỏ. Ta có thể mô phỏng giả định này thông qua đoạn mã trang sau. Chú ý là trong mô thức thông báo tập trung, một hộp hội thoại như vậy là không đơn giản vì luồng điều khiển nằm ngoài phạm vi của chương trình ứng dụng.

## Chương 8: Hỗ trợ cài đặt

lý do **repeat** phím bấm ôn lặp và hành động này sẽ diễn ra cho đến khi nào đó là số lần lặp mà bạn đã định sẵn. Ví dụ: **read-event (myevent)** là hành động bấm nút và **do** là lệnh để lặp lại cho đến khi nào đó là số lần lặp mà bạn đã định sẵn. Ví dụ: **case myevent.type** là một số lệnh mà bạn cần phải làm sau khi đã bấm nút.

**type-1: do type-1 processing**

**type-2: if(error-cond) then**

**repeat**

**read-event (myevent2)**

**case myevent2.type**

**type-1: do type-1 processing**

**type-n: do type-n processing**

**end case**

**until (end-cond)**

**end if**

**type-1: do type-1 processing**

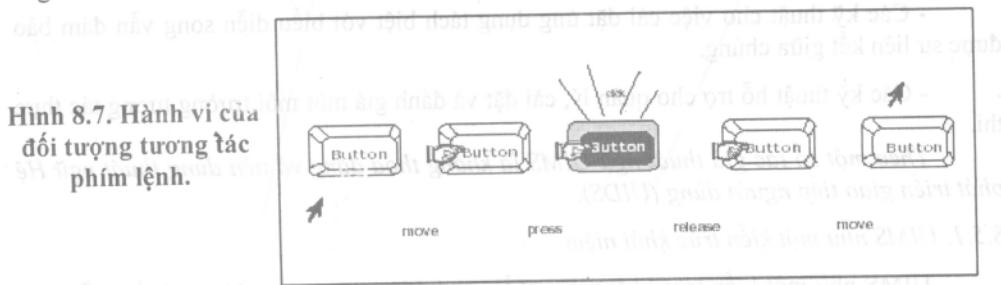
**type-n: do type-n processing**

**end case**

**until (end-cond)**

### 8.4. Sử dụng công cụ

Như chúng ta đã nêu trong chương ba, một đặc trưng quan trọng của giao tiếp chuẩn WIMP từ cảm nhận của người dùng đó là: trạng thái của đầu vào và đầu ra là hoàn toàn liên kết với các đối tượng độc lập trên màn hình hiển thị. Điều này tạo nên ảo giác là các đối tượng trên màn hình là các đối tượng được quan tâm, cái chúng ta gọi là *đối tượng tương tác* và điều này là cần thiết cho thế giới hành động của giao diện điều khiển trực tiếp. Hãy xét thí dụ kinh điển sau mà thiết bị trỏ ở đây là chuột. Đầu vào đến từ các thiết bị phần cứng được tách khỏi đầu ra của vị trí chuột trên màn hình hiển thị. Tuy nhiên, khi mà chuyển động trực quan của con trỏ màn hình được liên kết với chuyển động vật lý của thiết bị chuột, người dùng có cảm giác như chính mình đang điều khiển con trỏ trực quan. Thậm chí ngay cả khi đầu vào và đầu ra là tách biệt, cái ảo giác này gây cho người dùng phải xử lý chúng như một thay vì cả hai con trỏ trực quan và thiết bị vật lý cùng tham chiếu đến một cái là '*chuột*'. Trong tình huống nếu liên kết này bị đứt bể, dễ thấy sự chán nản của người dùng. Hình 8.7 minh họa việc tổ hợp vào ra cho tương tác dùng phím lệnh.



Hình 8.7. Hành vi của  
đối tượng tương tác  
phím lệnh.

## *Chương 8: Hỗ trợ cài đặt*

Trong sự cảm nhận của người dùng, sự kiện ở mức độ hệ thống window, đầu vào và đầu ra là còn khá tách biệt cho hầu hết trừ thiết bị chuột, và điều này cần cố gắng không ít trong chương trình ứng dụng nhằm tạo nên ảo giác về tương tác như kiểu tương tác phím lệnh ở trên.

Nhằm trợ giúp các nhà lập trình hợp hành vi đầu vào với đầu ra một mức độ trừu tượng khác được cung cấp đó là *Toolkits*. Toolkits cung cấp một tập các đối tượng tương tác định sẵn gọi là kỹ thuật tương tác với các widget, gadget. Các đối tượng tiền định này cung cấp một hành vi sẵn như đã miêu tả cho các phím lệnh ở trên và làm đơn giản cho việc lập trình. Toolkit tồn tại trong hầu hết các môi trường lập trình: X View cho X Window, Toolbox cho Macintosh, SDT cho MicroSoft Window.

Để tăng tính mềm dẻo, các đối tượng tương tác được điều chỉnh cho các tình huống cụ thể sẽ được triệu gọi bởi người lập trình. Thí dụ như chúng ta thấy trong nhiều ứng dụng, nhấn các phím lệnh có thể được tham số khi thì là nhấn văn bản, khi thì nhấn đồ họa tùy theo chủ ý của người lập trình.

Lập trình với Toolkit là tăng tính nhất quán và tính khái quát hoá như kiểu quan sát và cảm nhận (*look and feel*). Hơn nữa, việc sử dụng các đối tượng tương tác và toolkit làm cho ta gần với kỹ thuật lập trình hướng đối tượng. Chúng ta có thể sử dụng tính kế thừa đơn và kế thừa bội để việc lập trình đơn giản hơn. Thí dụ, để xây dựng các đối tượng tương tác phức tạp, ta có thể xây dựng từ các đối tượng tương tác đơn giản. Lớp có thể được định nghĩa như là "mẫu" cho các đối tượng tương tác. Và khi một đối tượng tương tác được tạo nó được khai báo như là một thể hiện của một vài lớp tiền định

### **8.5. Hệ quản trị giao tiếp người dùng (UIMS)**

Tuy đã cung cấp tính sẵn dùng và giá trị của các mức độ trừu tượng mà Toolkit đã thể hiện, cũng còn bức rào đáng kể mà người lập trình phải vượt qua trong quá trình thiết kế và cài đặt các hệ thống tương tác. Toolkit mới chỉ cung cấp một dài hạn chế các đối tượng tương tác và vẫn còn khó sử dụng với người không phải là lập trình. Như vậy, cần thiết phải bổ sung các hỗ trợ khác, đó là tập các chương trình và kỹ thuật nhằm hỗ trợ các mức độ dịch vụ khác cho việc thiết kế các hệ thống tương tác - các hệ thống quản trị giao tiếp người dùng - UIMS. Các mối quan tâm chính về UIMS là:

- Kiến trúc khái niệm đối với cấu trúc của một hệ thống tương tác, tập trung vào việc tách biệt giữa ngữ nghĩa và biểu diễn.

- Các kỹ thuật cho việc cài đặt ứng dụng tách biệt với biểu diễn song vẫn đảm bảo được sự liên kết giữa chúng.

- Các kỹ thuật hỗ trợ cho quản lý, cài đặt và đánh giá một môi trường tương tác thực thi.

*Theo một số tác giả thuật ngữ UIMS là không thoả đáng và nên dùng thuật ngữ Hệ phát triển giao tiếp người dùng (UIDS).*

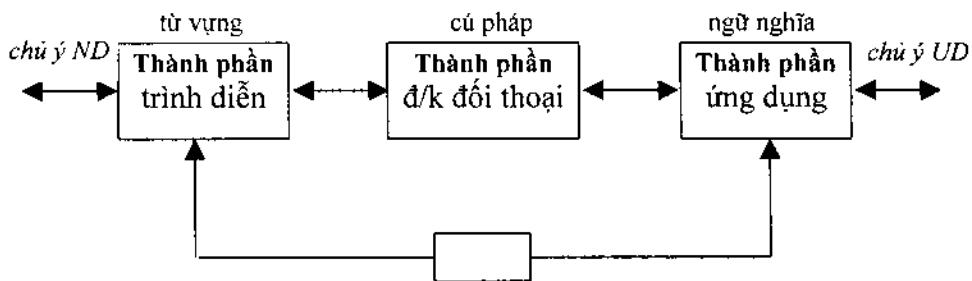
#### **8.5.1. UIMS như một kiến trúc khái niệm**

UIMS như một kiến trúc khái niệm nhằm tách biệt giữa ngữ nghĩa và biểu diễn ứng

## Chương 8: Hỗ trợ cài đặt

dụng. Các lý do chính, đó là: tính khả chuyên, tính sử dụng lại, đa giao tiếp, và tính cá nhân hoá.

Khi đã cho phép tách biệt ứng dụng với biểu diễn, chúng ta cần biết hai đối tác người dùng và ứng dụng giao tiếp với nhau ra sao. Vai trò của giao tiếp này được biết đến như là **điều khiển hội thoại**. Điều này cho phép chúng ta xem một hệ thống tương tác gồm 3 phần chính: **ứng dụng, trình diễn và điều khiển hội thoại** (hình 8.8).



Hình 8.8. Mô hình Seeheim về các thành phần lô gic của một UIMS.

Kiến trúc khái niệm đầu tiên của một UIMS đã được hình thành tại hội thảo 1985 ở Seeheim - Đức. Các thành phần logic của một UIMS như hình 8.8. Thành phần trình diễn chịu trách nhiệm về dáng vẻ của giao diện bao hàm cả đầu vào và đầu ra cho người dùng. Thành phần điều khiển đối thoại điều chỉnh giao tiếp giữa thành phần trình diễn và thành phần ứng dụng, còn giao tiếp ứng dụng là cách nhìn về ngữ nghĩa của ứng dụng được cung cấp như giao diện.

### 8.5.2. Các xem xét cài đặt

Dù đã phân biệt mô hình khái niệm với một số chú ý về cài đặt, song cũng cần thiết xem xét việc cài đặt các thành phần của một UIMS được thực hiện như thế nào? Việc cài đặt dựa vào mô hình Seeheim cần phải xác định sự tách biệt giữa các thành phần. Như trong phần 8.3, hệ thống window và các công cụ cung cấp việc tách biệt giữa ứng dụng và trình diễn. Việc sử dụng thủ tục *Callback* trong mô thức lập trình *Notifier* là một cách cài đặt, coi giao tiếp ứng dụng như một Notifier. Trong bộ công cụ chuẩn cho X Window, các thủ tục Callback này là trực tiếp, ứng dụng phải có trách nhiệm tự đăng ký với notifier. Trong mô hình MVC, các thủ tục Callback cũng được dùng để truyền thông giữa *Viewer* (V) hay *Controller* © với các mô hình liên kết của nó. Nhưng trong mô hình này, bộ phận của thành phần trình diễn (*Viewer* hay *Controller*) tự đăng ký với ứng dụng (mô hình). Nhiều kỹ thuật cho việc cài đặt tách biệt này đã đề cập ở một số chương trước, vì vậy trong phần dưới đây ta chỉ liệt kê một số.

#### Các kỹ thuật điều khiển đối thoại

- **Mạng menu:** Việc truyền thông giữa ứng dụng và trình diễn được mô hình hóa như mạng menu hay các menu con (submenu). Để điều khiển đối thoại, người lập trình chỉ đơn

## Chương 8: Hỗ trợ cài đặt

giản mã hoá mức của menu và sự liên kết của một menu với một submenu hay một hành động. Menu dùng để biểu hiện mọi điều vào có thể của người dùng tại bất cứ thời điểm nào. Chúng ta có thể coi một hệ thống kiểu Hypercard là thuộc dạng này.

- **Ký pháp văn phạm:** Việc đối thoại giữa ứng dụng và trình diễn được xử lý như một văn phạm hành động và trả lời và thường dùng văn phạm phi ngữ cảnh như BNF. Kỹ thuật này khá tốt với kiểu giao tiếp dòng lệnh, nhưng không tốt lắm cho một số giao tiếp đồ họa. Nó thực sự khó khăn khi phải mô hình hóa giao tiếp của các giá trị chéo qua điều khiển đối thoại và cần thiết phải duy trì một vài ngữ nghĩa phản hồi từ ứng dụng đến trình diễn.

- **Lưu đồ chuyển trạng thái:** Lưu đồ trạng thái có thể dùng như một công cụ biểu diễn đối thoại. Nhiều biến thể của lưu đồ dịch chuyển trạng thái đã đề cập trong chương trước. Tuy nhiên cái khó của nó là việc liên kết các sự kiện của đối thoại với việc biểu diễn tương ứng hay sự kiện của ứng dụng. Hơn nữa, sự giao tiếp giữa ứng dụng và trình diễn không thể biểu diễn rõ ràng.

- **Ngôn ngữ sự kiện:** Ngôn ngữ sự kiện tương tự như ký pháp ngôn ngữ, song nó có thể được thay đổi để biểu diễn một cách trực tiếp và hỗ trợ ngữ nghĩa cho thông tin phản hồi. Ngôn ngữ sự kiện tốt cho việc miêu tả hành vi vào ra được cục bộ theo nghĩa của luật sản xuất.

- **Ngôn ngữ mô tả:** Tất cả các kỹ thuật trên, trừ mạng menu, đều không tốt cho việc miêu tả sự tương ứng giữa ứng dụng và trình diễn bởi vì chúng không có khả năng miêu tả một cách hiệu quả luồng thông tin giữa hai thành phần như thế nào? Chúng chỉ xem đối thoại như một chuỗi sự kiện xảy ra giữa hai đối tác. Cách tiếp cận đặc tả tập trung vào việc miêu tả mối liên hệ giữa ứng dụng và trình diễn. Mỗi quan hệ này có thể xem như là một CSDL các giá trị dùng chung cho cả hai đều có thể truy nhập.

- **Ràng buộc:** Hệ thống ràng buộc là một tập con đặc biệt của các ngôn ngữ mô tả. Các ràng buộc có thể sử dụng để thiết lập một cách tường minh mối liên kết giữa thông tin độc lập của trình diễn với ứng dụng. Kỹ thuật này được sử dụng trong mô hình PAC hay AVL. Ràng buộc biểu hiện sự phụ thuộc giữa các giá trị khác nhau cần phải được duy trì. Thí dụ một con heo đất thông minh phải hiện số tiền mà nó chưa, và ràng buộc ở đây là giá trị nó hiện ra bên ngoài phải giống như số tiền nó chưa bên trong.

### - Đặc tả đồ họa:

Khuynh hướng của điều khiển đối thoại

- Điều khiển trong (vòng lặp đọc - đánh giá)
- Điều khiển ngoài, thí dụ như tính độc lập ngữ nghĩa của ứng dụng hoặc biểu diễn
- Điều khiển biểu diễn như các đặc tả đồ họa.

# 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 HC:**

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 hóa 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 này 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 tài 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 chuyên 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. Điều 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

Phản ứng 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 đó. 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 đắn. Thường, điểm nhấn chính của kỹ thuật này là thiết lập tính dễ học và dễ 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 già định.

Với thông tin trên, người đánh giá thực hiện chuỗi các hành động và đưa ra các ý kiến về tính đúng đắn. 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 khảng 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 có 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 (dịch)
2. Các vấn đề cho hiệu chỉnh mục tiêu
  
- 
- 
- 
3. Vấn đề xác định hành động
  
- 
- 
- 
4. Vấn đề thực hiện hành động
  
- 
- 

Để 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):

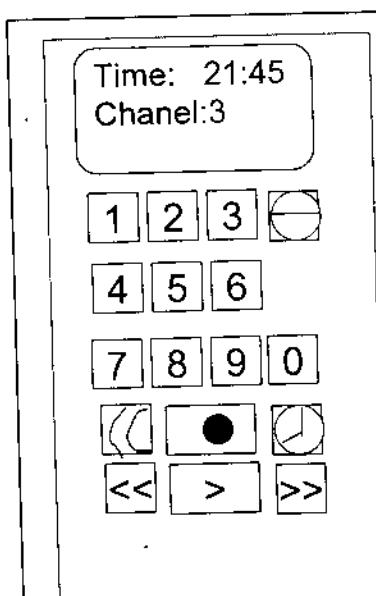
*Action A: Nhấn phím thời gian ghi*

*Đáp ứng A: Chuyển về hiển thị theo mode thời gian. Con trỏ xuất hiện sau chữ Start.*

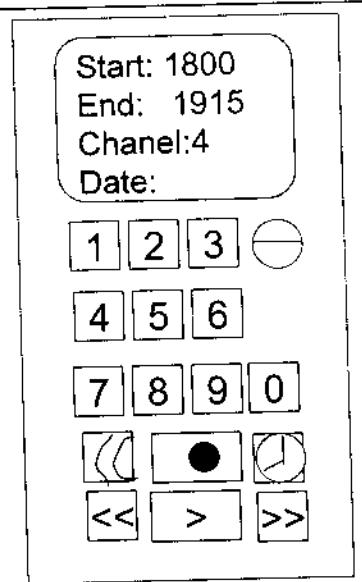
## Chương 9: Các kỹ thuật đánh giá giao tiếp người dùng

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.



Hình 9.1a. Trạng thái ban đầu



Hình 9.1b. Nhấn phím ghi

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 đúng đắn 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 đúng đắn. Ý tưởng chính ở đây là nhiều người đánh giá độc lập cùng tiến hành trên một hệ thống để nêu lên tính đúng đắn. Kỹ thuật này tốt nhất nên sử dụng trong giai đoạn đầu của thiết kế vì nó cố định nhiều vấn đề của tính đúng đắn. 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, Nielsen 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.

**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 lối tốt**

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

**9) Dự báo lối**

**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 lối

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 phần 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 Keytrack 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
- Giả 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:  $T_M + T_K(File) + T_P + T_K(Save)$
  - 4) MS Word nhắc nhập tên tệp:  $T_R$ ; người dùng nhấn "File-2.4", cuối cùng nhấn phím Enter:  $T_K(return)$ :  
Tổng thời gian thực hiện:  $T_R + T_M + T_K(File-2.4) + T_K(return)$

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 đề 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 đề 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 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 thứ 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.

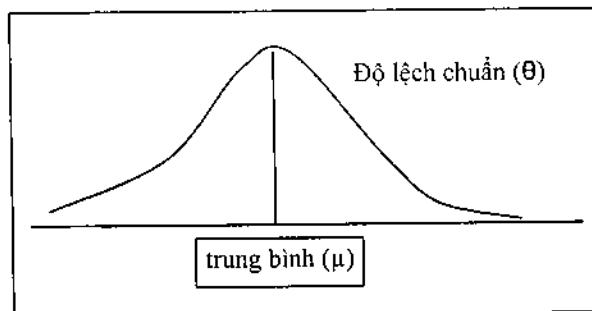
Khi chúng ta đã xác định xong giả thiết mà chúng ta sẽ cố gắng đánh giá, khi các biến đã được nghiên cứu, các chủ đề đã bố trí xong, và thiết kế là phù hợp, việc còn lại là quyết định sẽ phân tích kết quả thu được như thế nào? Phương pháp phân tích thống kê có vẻ thích hợp và có nhiều phương pháp có 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.

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, ví dụ như màu màn hình: đỏ, xanh, trắng,... Biến liên tục có thể nhận vô số giá trị, ví 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 tuỳ 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, do 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 có 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 thấu đáo 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 mồi 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..

### **Thể thức phân tích kết quả**

Có nhiều cách ghi lại hành động người dùng :

i) Bút và giấy: Đây là phương pháp rất nguyên thuỷ, 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 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 thể 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 dàng 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 thể thức tự động: Thể 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 chủ thích cho việc quan sát, quay video hay ghi nhật ký và đồng bộ hóa 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 rẻ đối với người quản trị. Tuy nhiên, thông tin thu được cần phải có chủ 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 tồn 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, để 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 mới 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 kiêu 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, ví 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 hiệu được khuynh hướng của người dùng hệ thống (nếu đó là một hệ thống khai thác) 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.

**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 lối 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.

Vì 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 Ginevra, Thụy Sĩ (European Laboratory for Particle Physics in Geneva) do Tim 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, Nescape, 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,...

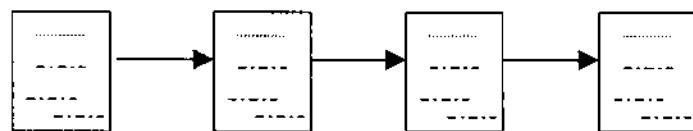
#### 10.1.1. Siêu văn bản, siêu liên kết, multimedia

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 mươi năm trước thời những thuật ngữ này có lẽ còn chưa xuất hiện chứ không muốn nói đến khái niệm.

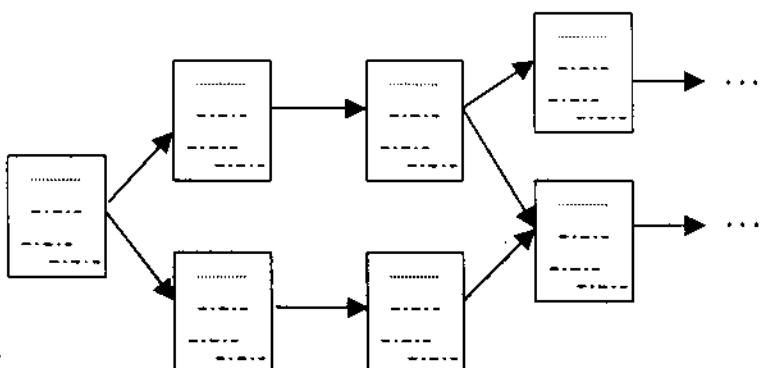
Khái niệm Siêu văn bản do nhà tin học Ted Nielson đưa ra năm 1965 trong bài viết “Computer Dreams”, bàn về ý tưởng tạo ra các máy tính có tư duy như con người. Nhưng người thực hiện ý tưởng ấy lại là Tim Berners Lee vào khoảng những năm 80 và đến năm 1989, Tim và các cộng sự của mình đã thực hiện thành 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 thi 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.

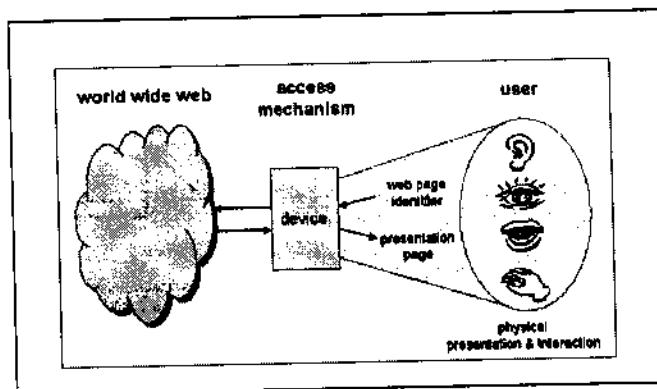
Chú ý rằng đa phương tiện hay siêu phương tiện không chỉ dành cho ứng dụng Web mà còn trong nhiều các ứng dụng khác như xây dựng các công cụ cho quá trình phát triển phần mềm (tạo mẫu thử, đánh giá thiết kế,...), các hệ thống đào tạo từ xa, v.v. Tuy rằng, đây không phải điều quan trọng chính của chúng ta, song có lẽ tốt hơn nếu chúng ta điểm 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 diển 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, anh 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 tài liệu liên quan trên Internet. Các trang Web ngoài việc hiển thị 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 đồ vỡ (Linkrot), mà điều này cũng rất khó tránh. Các thống kê cho rằng có 0,6% kết nối bị đồ 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 đến 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 vào 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 ẩn 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 sĩ 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 Nielson và một số các chuyên gia, có 10 lỗi phổ biến thường gặp khi thiết kế các trang Web [10]:

1. Thời gian tài 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 chảy 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 tài
4. Các kỹ thuật HTML để thể hiện giao diện
5. Thao tác giữa các phần hay môi trường người dùng

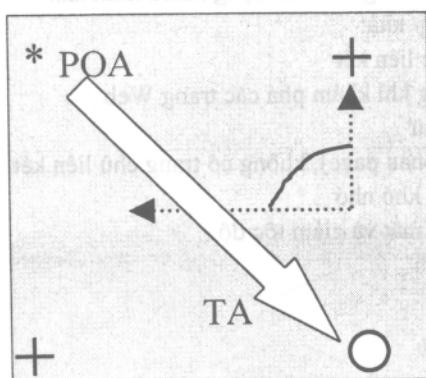
#### 10.3.2.1. Thiết kế trực quan

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.

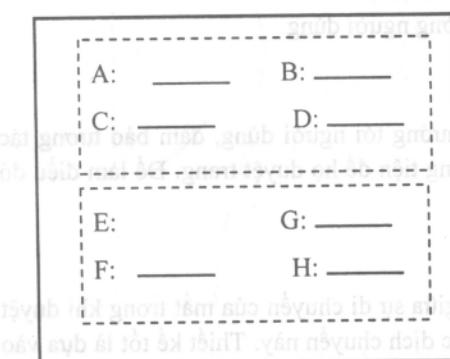
Chú ý rằng, con người có thói quen đọc từ trái qua phải và từ trên xuống dưới. Có thể hình dung điều này thông qua sơ đồ Guternberg trong hình 10.3 dưới đây.



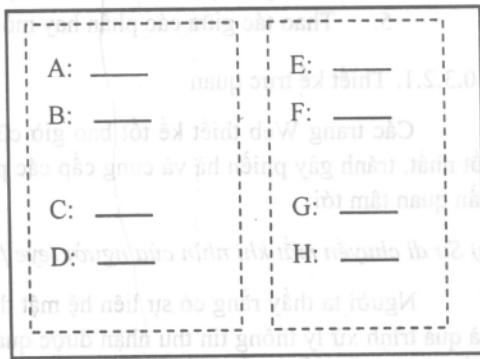
- POA (Primary Optical Area): điểm xuất phát.
- TA (terminal area): điểm kết thúc.
- Đường chấm gạch: thể hiện hướng đưa mắt ngược với hướng đưa mắt của người.
- Dấu '+': đánh dấu biên.

Hình 10.3. Lưu đồ Gutternberg.

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.

The screenshot shows a Yahoo! Mail sign-up form with a grid layout. The columns include:

- Choosing your ID**: You will use this information to access Yahoo! Mail each time. Capitalization matters. Only lowercase letters are allowed.
- Sign up for your Yahoo! ID with Mail**: Get a Yahoo! ID and password for access to Yahoo! Mail and all other personalized Yahoo! services.
- Help - Yahoo!**
- Already have an ID? Sign In**
- Mail Out To:** [Email Address] @ yahoo.com
- Password:** [Password]
- Re-type Password:** [Re-type Password]
- If you forget your password, we will only ask you for this information.**
- Security Question:** [Select a question] Your Answer: [Text input]
- Birth day:** [Day] / [Month] / [Year]
- Current Email:** [Text input]
- Recalling your password**: This is our only way to verify your identity. To protect you, we'll make sure your "password" 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 lượt 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).

## Chương 10: Multimedia và Web

Hình 10.6. Cấu trúc phân cấp trực quan.

Hình 10.7. Cấu trúc phân cấp trực quan.

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><b>Centred text alignment is the mark of an amateur.</b></p> <p>Do you find this easy to read?</p>	<p><b>Right alignment, or 'ragged left' as it is sometimes called</b></p> <p>is though to be quite tremdy.</p> <p>What do you think about its impact on readability?</p>
<p><b>Left alignment</b> 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>	<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>

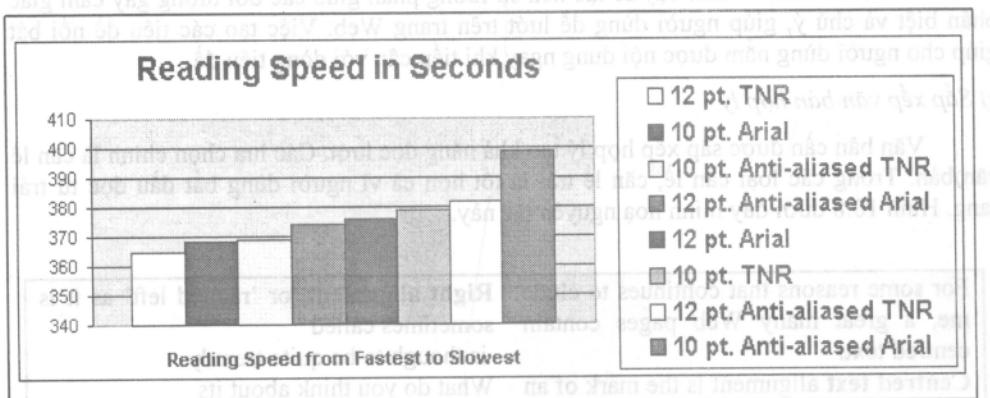
**Hình 10.8. Căn lề văn bản trên trang Web.**

d) *Tính dễ xem (Legibility)*

Các nhân tố tạo nên tính dễ xem bao gồm: cỡ chữ, phông chữ, màu sắc và sự tương phản. Các nghiên cứu trong phòng thí nghiệm (Software Usability Research Lab Wuchina 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 khuyên cáo là tránh sử dụng toàn chữ hoa, hay dùng chữ nháy hoặc chuyển động kiểu marque 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 Nielsel 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 Nielsel 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 tạo cho người dùng cảm giác毳, 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 (vẫn đề độ trễ)
- 10 giây để tạo được sự chú ý của người dùng. Độ 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? Đầu tiên, 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 cỗ 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 họa 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, Amazon 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 giãn 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? Thủ nghiêm 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 án đóng 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)**

Mỗi 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 mờ 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 môi 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

thay đổi 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, thậtẠI 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 quyền sách hay thư viện, chứ đừng chọn tàu vũ trụ hay vô 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

òn định. Tính ổn định chức năng có hai thành phần - đặt các vật dụng 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.

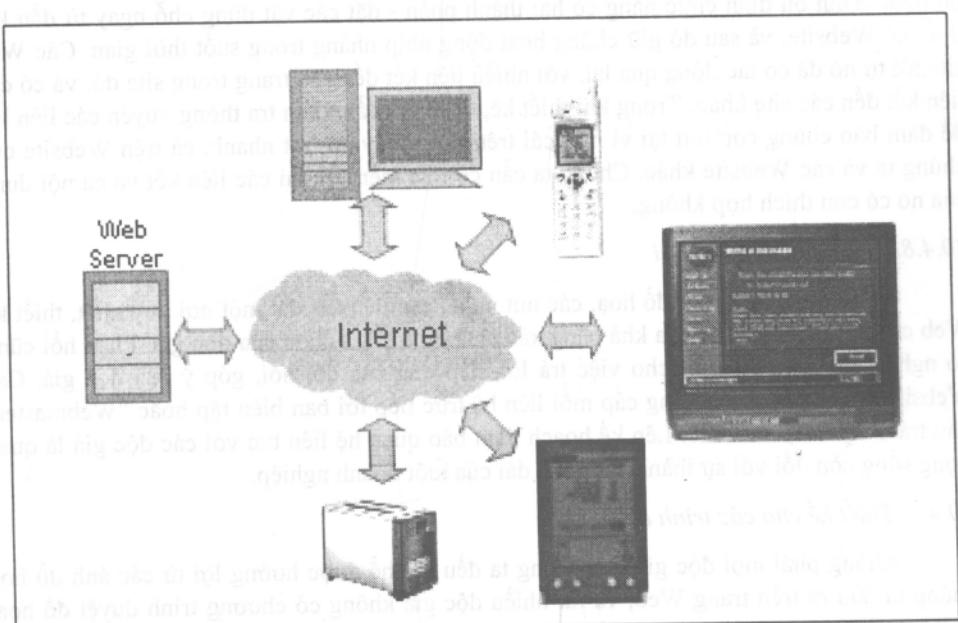
#### *10.4.10. Chủ ý cuối cùng*

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 bức bối.

### **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, tivi, 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)** là giao diện người dùng có khả năng tương tác với người dùng bằng cách kết hợp các phương thức khác nhau như giọng nói, hình ảnh, video, văn bản, v.v.

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 mắt và tay được sử dụng cho các việc khác.

#### *10.5.5. Người mùi 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 người 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 kế 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 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 trong 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 điểm 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 phù 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 bắt chấp sự thành thạo của một số người đang tìm kiếm trợ giúp hay tinh 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 ngắt công việc của người sử dụng. Một giải pháp khả thi là chia đôi của số 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).

```

  Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Kaka>help md
Creates a directory.

MKDIR [drive:]path
MD [drive:]path

If Command Extensions are enabled MKDIR changes as follows:

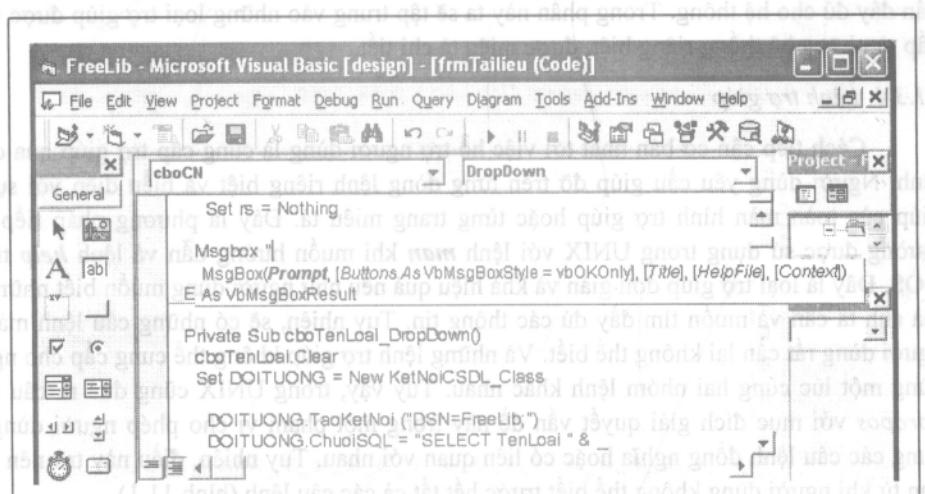
MKDIR creates any intermediate directories in the path, if needed.
For example, assume \a does not exist then:
      mkdir \a\b\c\d
is the same as:
      mkdir \a
      chdir \a
      mkdir b
      chdir b
      mkdir c
      chdir c
      mkdir d

which is what you would have to type if extensions were disabled.
  
```

**Hình 11.1.** Trợ giúp dòng lệnh trong DOS.

11.3.2. Dấu nhắc lệnh

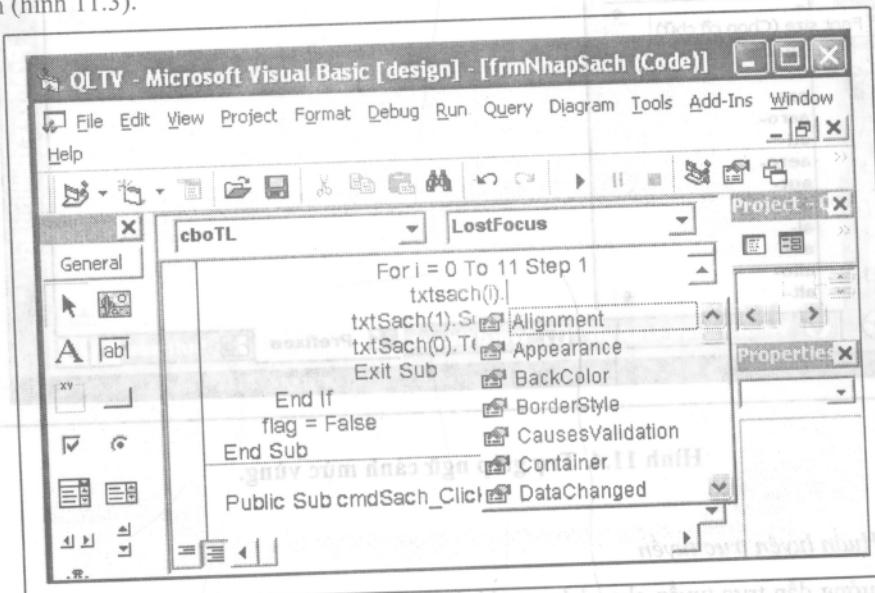
Dấu nhắc lệnh cung cấp sự trợ giúp khi người dùng bắt gặp một lỗi, thường ở trong dạng dấu nhắc sửa lỗi. Mỗi lời nhắc là khá hữu ích nếu như đó là lỗi đơn, như lỗi cú pháp. Hoặc là khi hướng dẫn cho người sử dụng cú pháp của câu lệnh, điều này được thể hiện khá rõ trong môi trường lập trình.



**Hình 11.2.** Một thí dụ về trợ giúp trong Visual Basic.

## Chương 11: Thiết kế trợ giúp

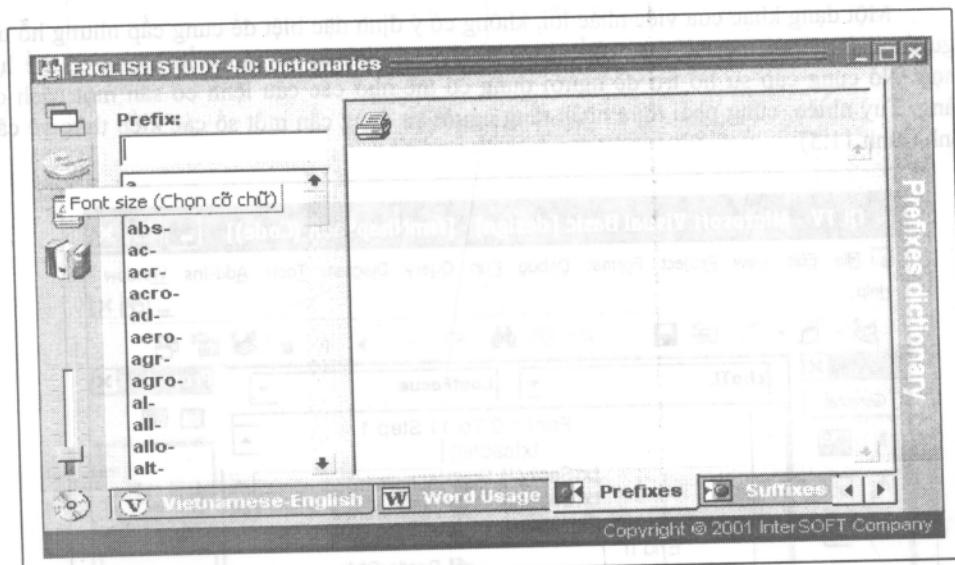
Một dạng khác của việc nhắc lỗi, không có ý định đặc biệt để cung cấp nhưng hỗ trợ người sử dụng với mức độ hạn chế, được sử dụng là *Menu* và các biểu tượng có thể lựa chọn. Nó cung cấp sự hỗ trợ để người dùng có thể nhớ các câu lệnh có sẵn một cách dễ dàng. Tuy nhiên, cũng phải thừa nhận rằng người sử dụng cần một số kiến thức về câu lệnh (hình 11.3).



Hình 11.3. Trợ giúp kiểu menu của VB.

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 mức vùng. Chúng thường được sử dụng trong các hệ thống dựa menu để cung cấp sự 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.



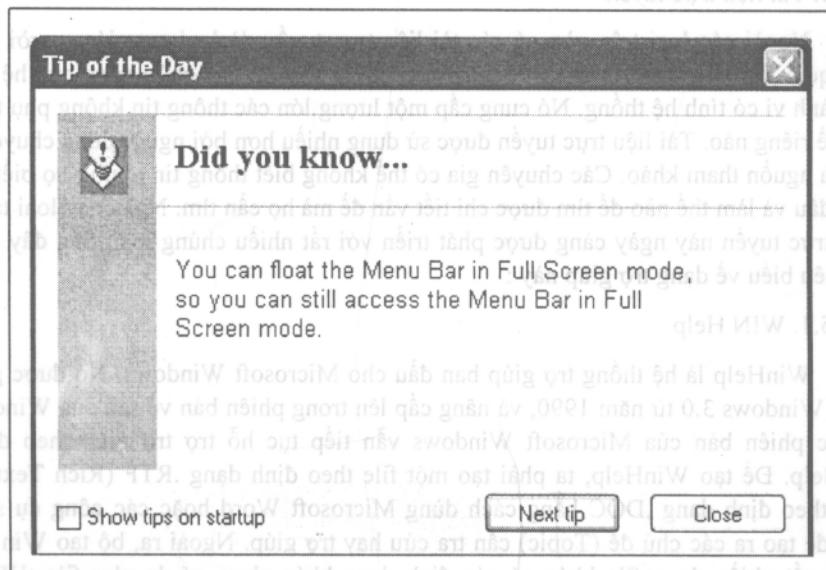
Hình 11.4. Trợ giúp ngữ cảnh mức vùng.

#### 11.3.4. Huấn luyện trực tuyến

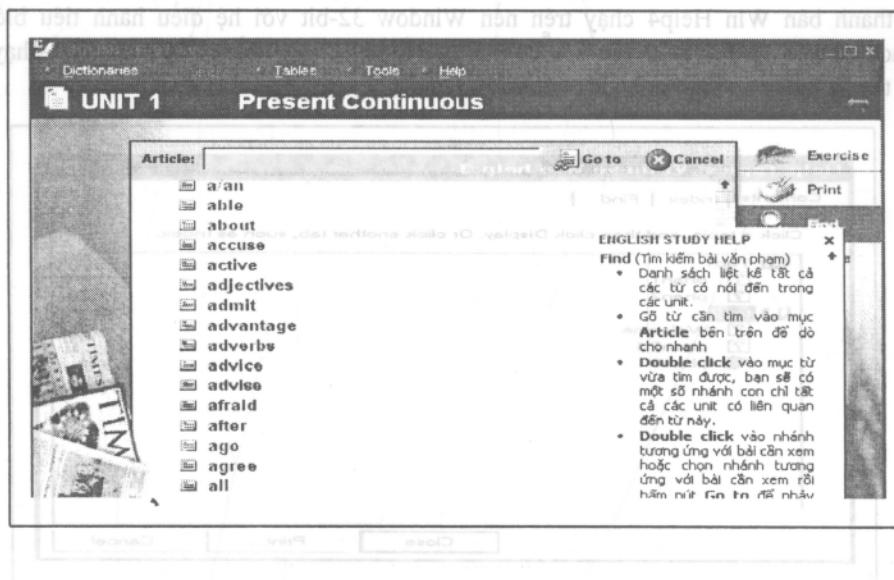
Hướng dẫn trực tuyến cho phép người dùng làm việc với những trợ giúp ngay trong ứng dụng. Người sử dụng có thể tiến hành tra cứu và lặp lại phần đó nếu thấy cần. Anh ta sẽ cảm nhận được công việc bằng việc thực hành với các ví dụ. Hầu hết các hướng dẫn trực tuyến không “thông minh”: chúng không biết 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, có 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 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).

## Chương 11: Thiết kế trợ giúp



Hình 11.5. Một dạng trợ giúp trực tuyến.



Hình 11.6. Một phiên bản của trợ giúp trực tuyến.

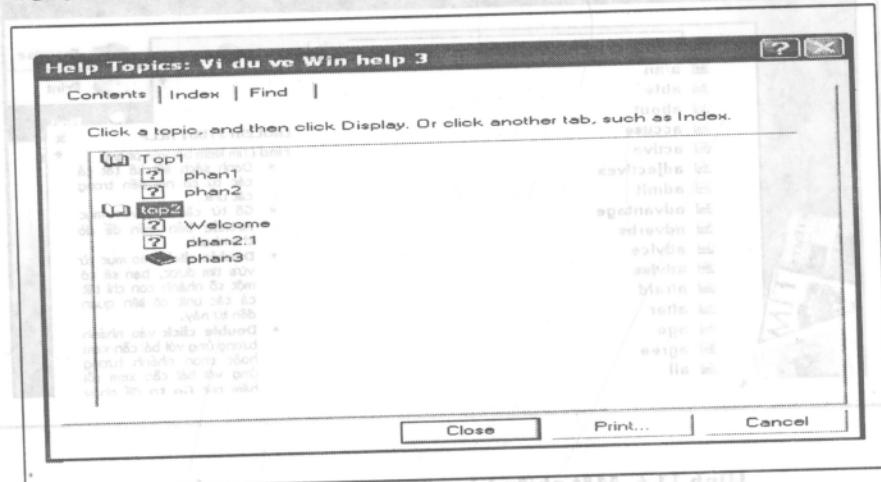
## Chương 11: Thiết kế trợ giúp

### 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 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 file 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.



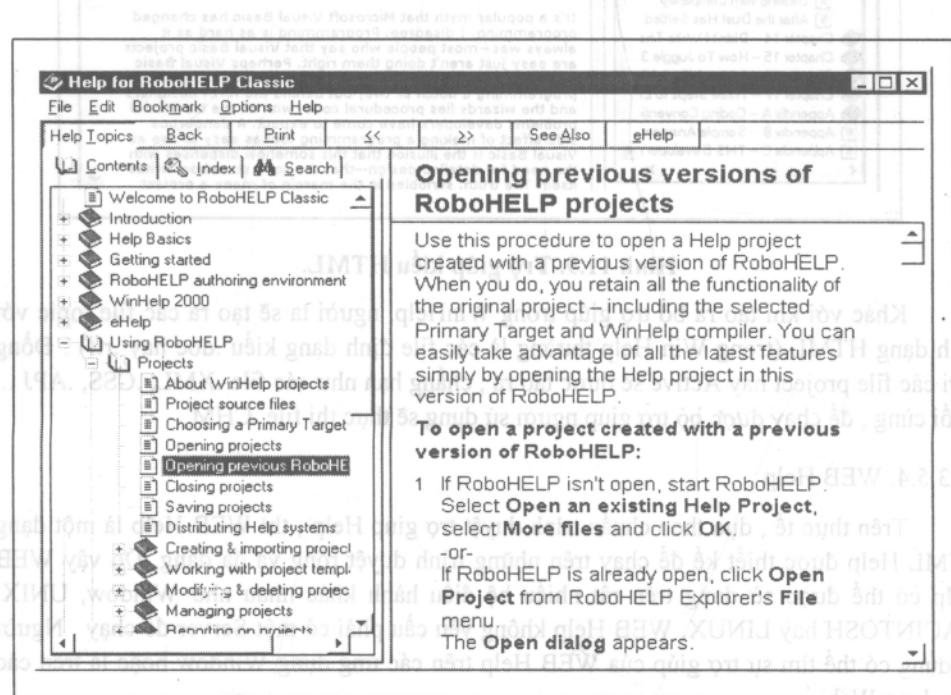
Hình 11.7 Trợ giúp của WINHELP

## 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

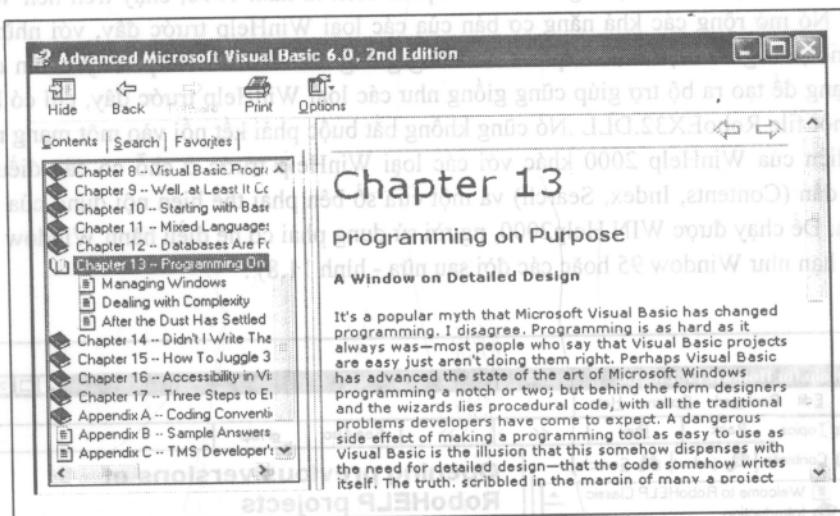
WinHelp 2000 được nghiên cứu và phát triển từ năm 1998, chạy trên nền Window 32-bit. Nó mở rộng các khả năng cơ bản của các loại WinHelp trước đây, với những tính năng mới, có giao diện như Explore và trông giống như HTML Help. Tuy nhiên các file định dạng để tạo ra bộ trợ giúp cũng giống như các loại 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.

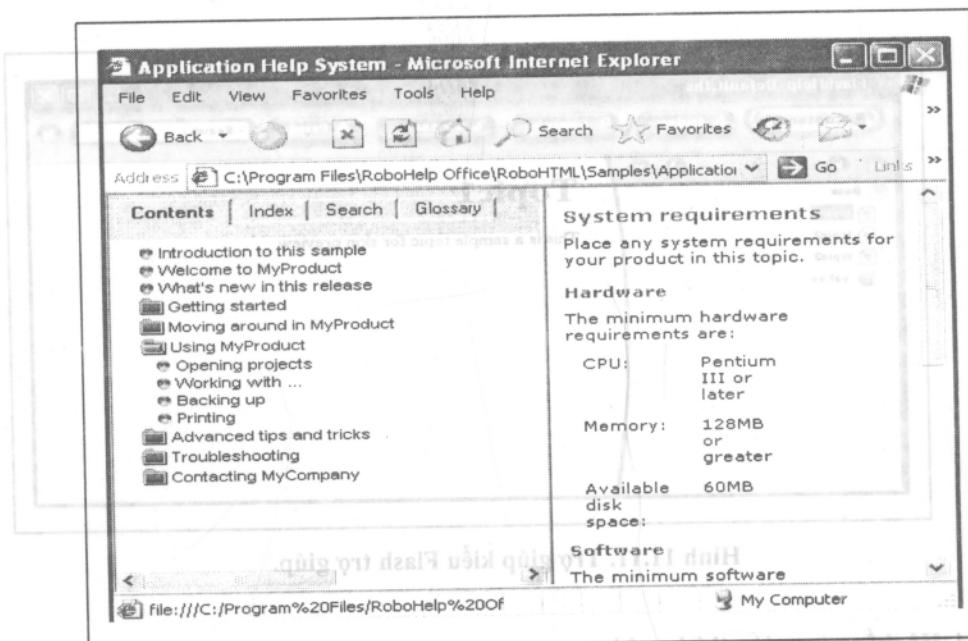
### 11.3.5.4. WEB Help

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 bố 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.



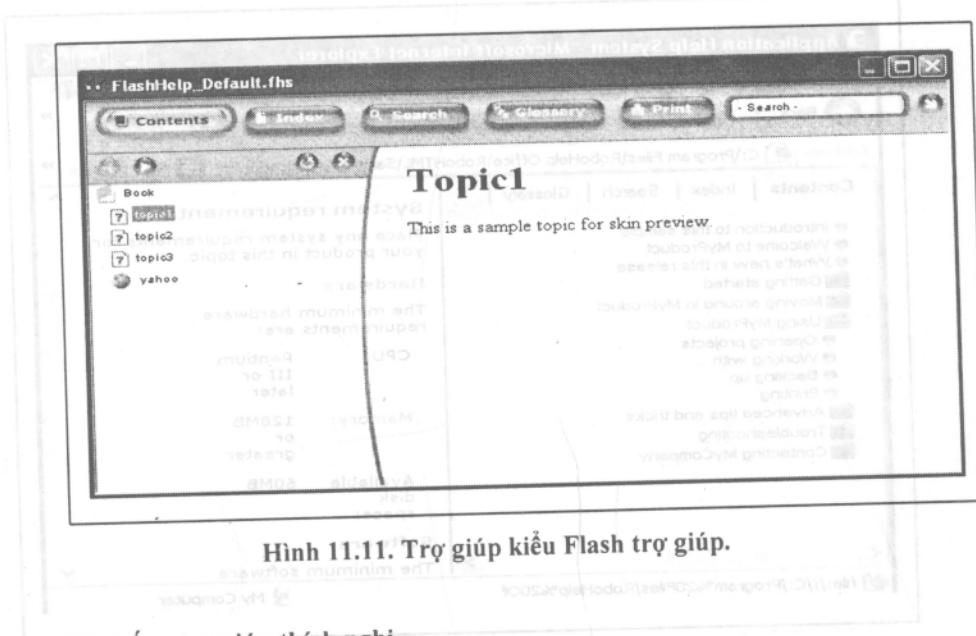
Ngoài ra còn có một dạng WEB Help nữa đó là dạng WEB Help Pro. Đây là một dạng Help server-based được thiết kế để hiển thị trong tất cả các trình duyệt lớn cũng như các nền tảng khác nhau như là WEB Help. Tuy nhiên các thành phần của WEB Help Pro có thêm các tính năng hết sức mạnh mẽ. Chẳng hạn như nó 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

**Mô tả:** 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 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

vẫn 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 tri 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 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 do 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 phần 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 săn:** Một cách tiếp cận khác trong việc mô hình hoá người dùng tự động là dùng mẫu săn. 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 săn. *Những mẫu săn 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 săn. 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 phủ: Đây là một mô hình được lý tưởng hóa, 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 hóa 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 hóa 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 thịnh thoảng thuộc về trợ giúp thích ứng là các chiến thuật cỗ vấn hoặc hướng dẫn mô hình hoá cung cấp một hệ thống trợ giúp với kiều 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 kiều 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 kiều 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 kiều 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 kiều cách vai kề 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 file  
And  
last command is COMPILE file  
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 gán 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 gõ rồi 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 thẻ hiện (của) sự COMPILE*

*COMPILE là một lệnh*

*COMPILE có liên quan đến DEBUG*

*COMPILE có liên quan đến EDIT*

*Bộ gõ lỗi tự động làm đơn giản việc Gõ 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 trợ giúp

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 tối 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 mối quan tâm để suy ra những thử nghiệm 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 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 cần 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. Hỗn 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 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 II: 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 chỉ 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 chóng hơn một hệ thống tĩnh vì mất những nhiêu 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 tắt yêu cầu 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 nêu 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.

## **Chương 11: Thiết kế trợ giúp**

---

### **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, sau đó 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

thước các cửa sổ Help vì chúng che phủ lên màn hình ứng dụng.

- Khi sử dụng các cửa sổ Help cảm ngữ 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 dù 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 cẩn lè 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

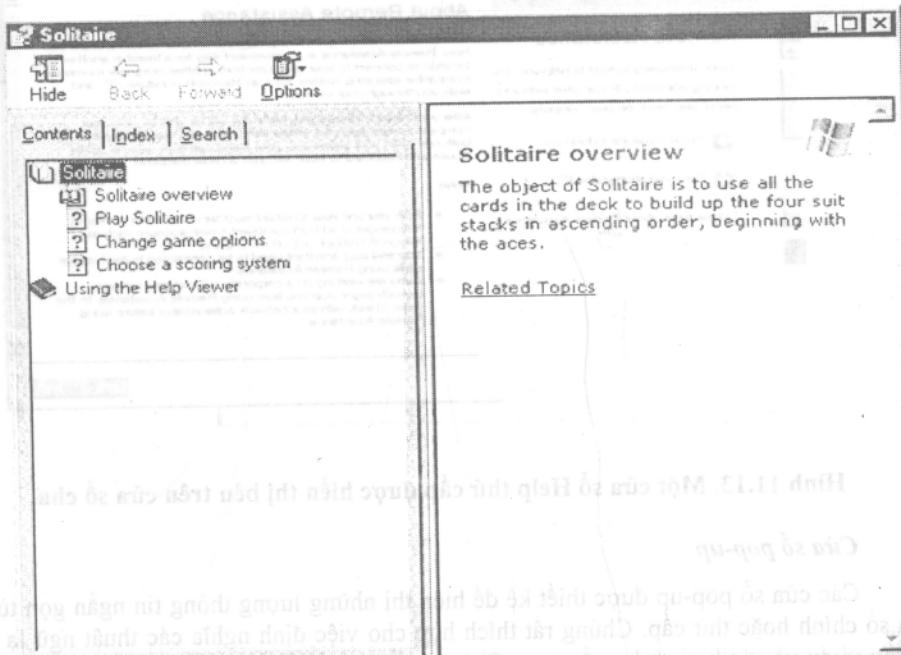
Có ba loại cửa sổ trong HTML Help: các cửa sổ chính, cửa sổ thứ cấp, và cửa sổ pop-up. Các cửa sổ chính và cửa sổ thứ cấp được biết tới như các cửa sổ tĩnh. Chúng vẫn tồn tại trên màn hình của người dùng cho đến khi họ đóng lại. Các cửa sổ thứ cấp mà xuất hiện từ các liên kết trong cửa sổ chính sẽ tự 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.



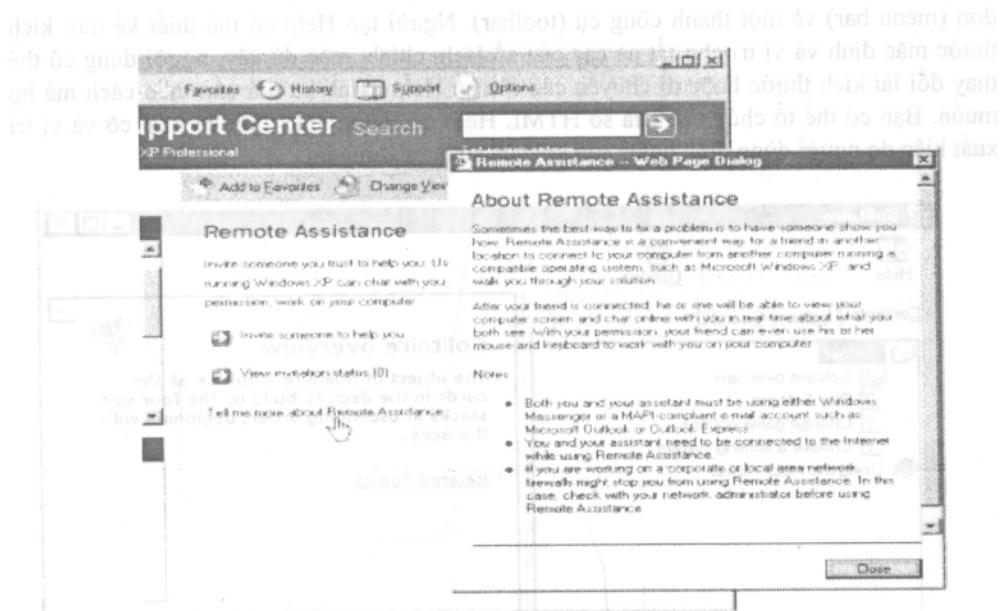
Hình 11.12. Một cửa sổ Help chính.

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

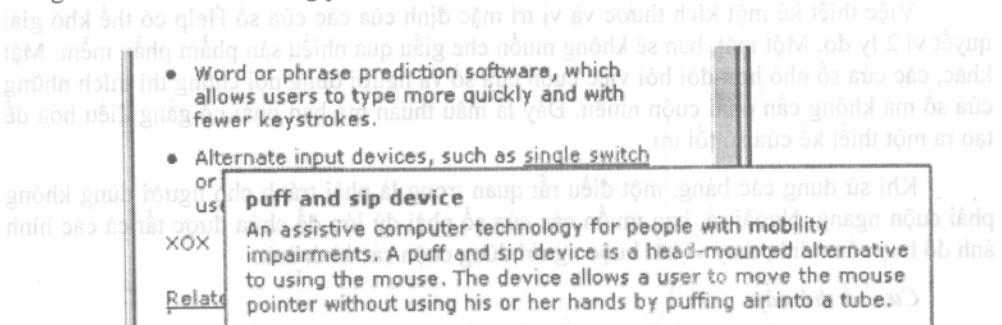
Cửa sổ thứ cấp xuất hiện cùng hoặc thay thế cho một cửa sổ Help chính. Bạn có thể sử dụng chúng cho nhiều mục đích. Tốt nhất là nên thiết kế các cửa sổ thứ cấp sao cho chúng không bao trùm lên phần ứng dụng mà chúng giải thích.



Hình 11.13. Một cửa sổ Help thứ cấp được hiển thị bên trên cửa sổ cha.

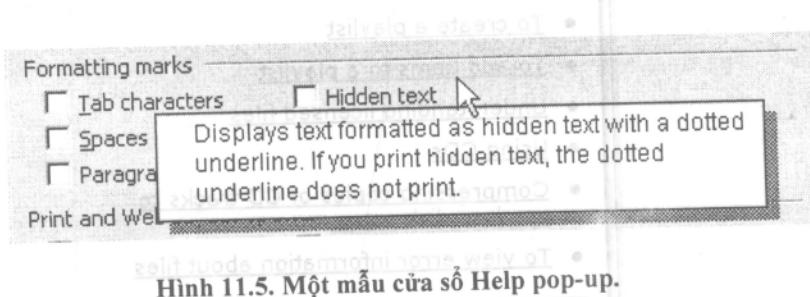
#### Cửa sổ pop-up

Các cửa sổ pop-up được thiết kế để hiện 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 lẻ 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.



Hình 11.14. Cửa sổ pop-up.

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:**

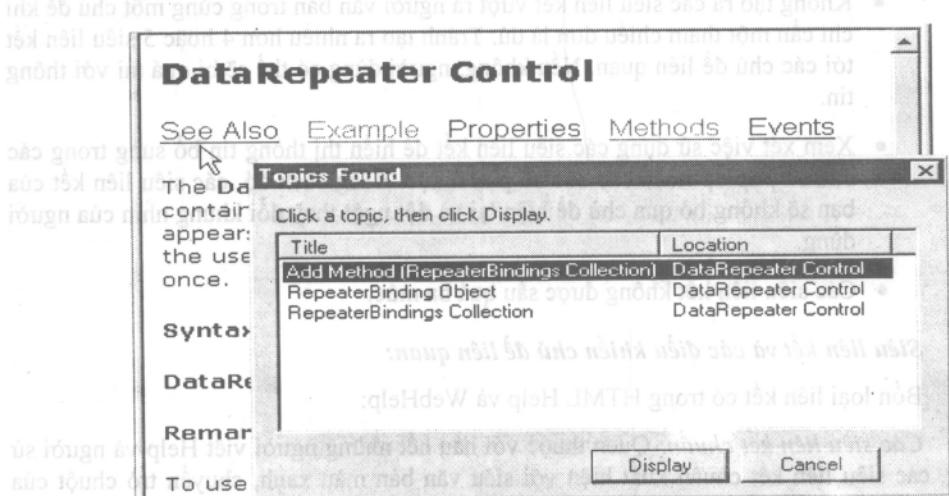
Bốn loại liên kết có trong HTML Help và WebHelp:

**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.

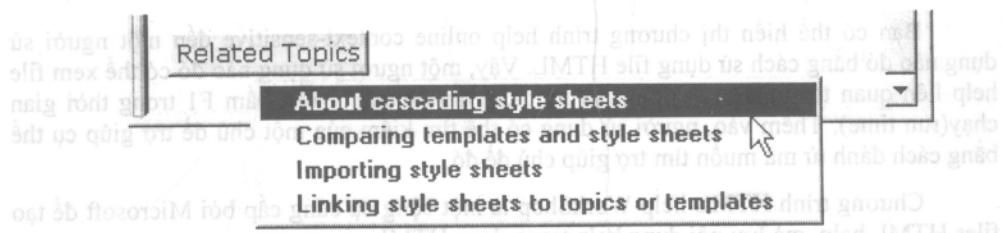
Các điều khiển See Also: Thay vì việc tạo ra rất nhiều các siêu liên kết để truy nhập tới các chủ đề liên quan, người viết Help có thể sử dụng một nút “See Also” hoặc một liên kết để chỉ ra một danh sách các chủ đề liên quan trong một hộp thoại hoặc một cửa sổ pop-up. Danh sách các chủ đề được tạo ra 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 dưa 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 xoá 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 xoá 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 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.

## **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 đê:

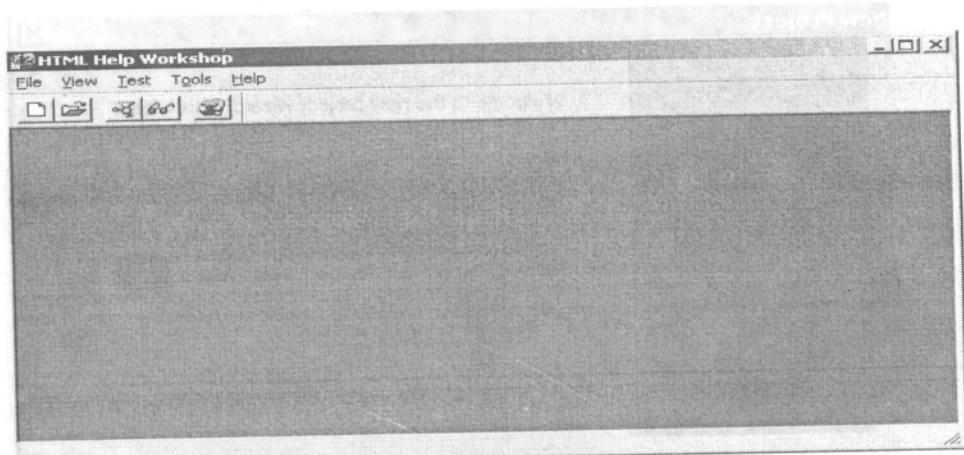
- Thêm vào chi tiết của một khách hàng mới.
- Sửa đổi chi tiết của một khách hàng tồn tại trong hệ thống.
- Xoá chi tiết về một khách hàng tồn tại
- Điều hướng qua các chi tiết của khách hàng tồn tại.

#### **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:

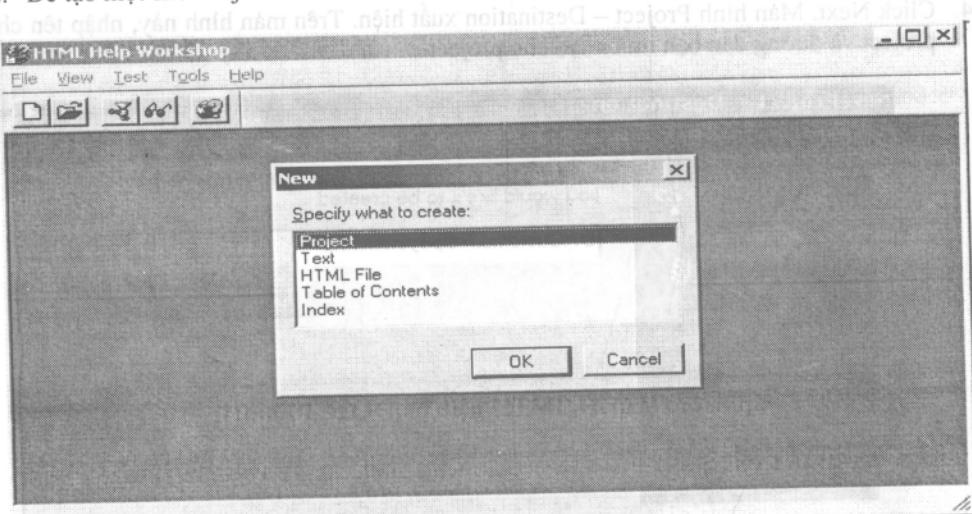
## *Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop*



**Hình PLA.1. Giao diện HTMP Help Workshop.**

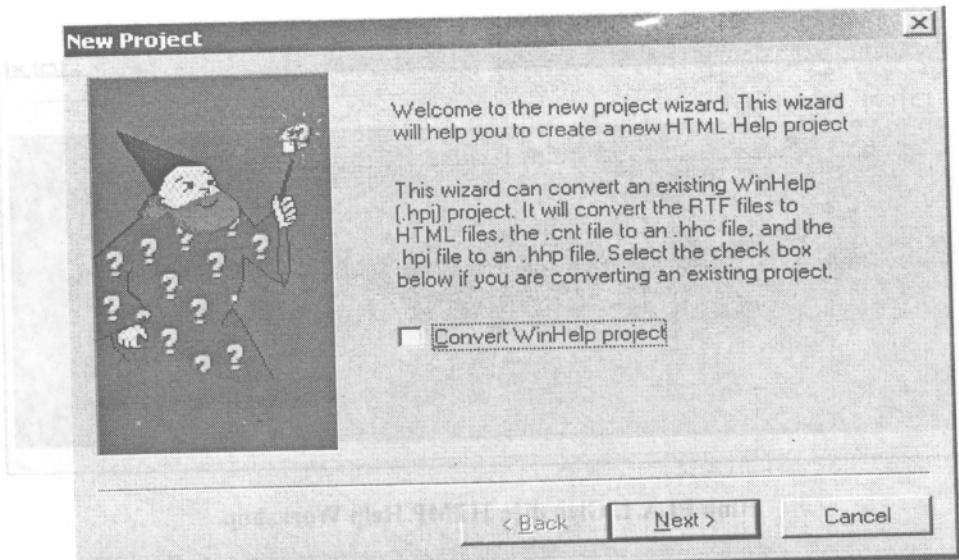
Một hệ thống trợ giúp được biên dịch bởi một file Project. Một file project chứa tất cả các trang chủ đề đã tạo ra của một hệ thống trợ giúp.

- Để tạo một file Project từ thanh thực đơn: File → New → New dialog box xuất hiện



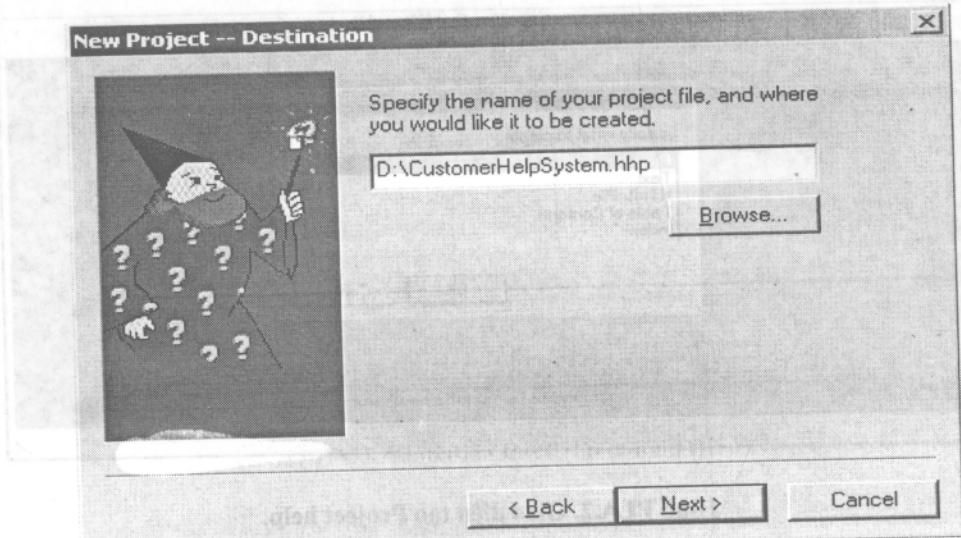
**Hình PLA.2. Giao diện tạo Project help.**

- Lựa chọn Project được lựa chọn theo mặc định. Click OK.



**Hình PLA.3. Giao diện tạo Project help (tiếp).**

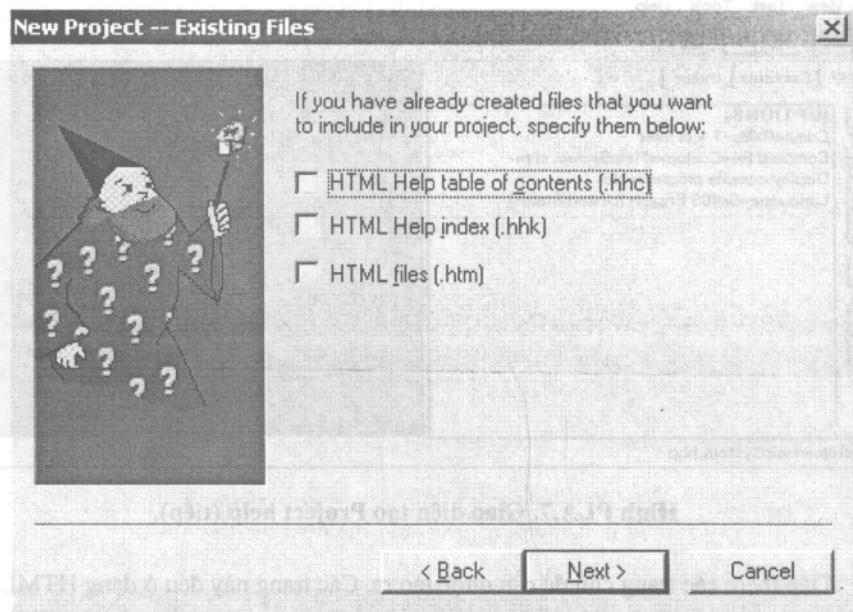
4. Click Next. Màn hình Project – Destination xuất hiện. Trên màn hình này, nhập tên cho project và đường dẫn bạn muốn tạo cho project.



**Hình PLA.4. Giao diện tạo Project help (tiếp).**

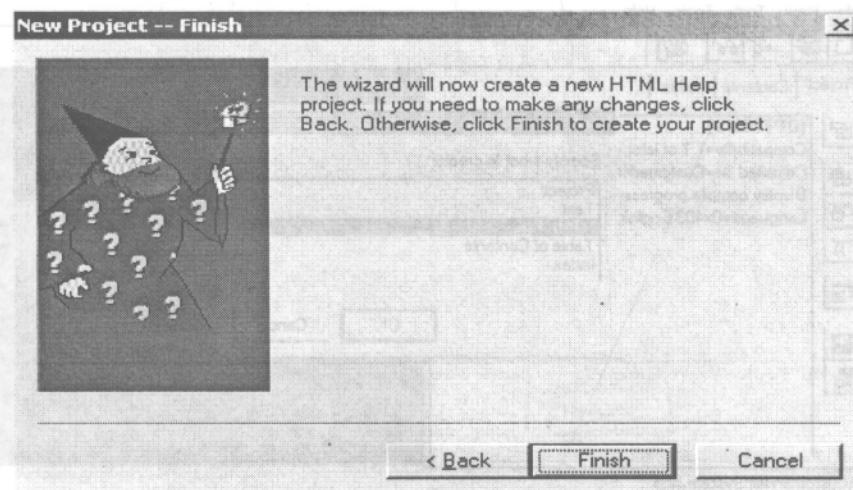
## **Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop**

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).**

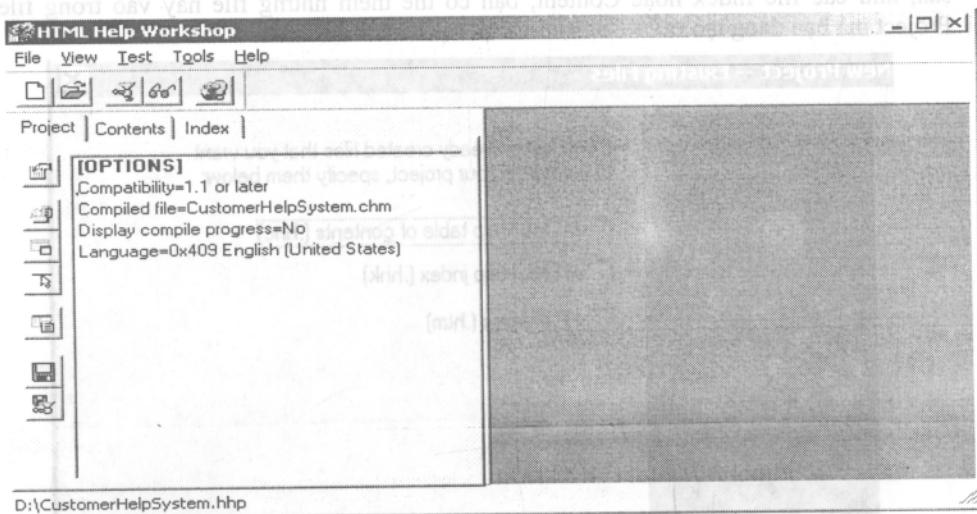
6. Màn hình New Project - Finish xuất hiện. Click Finish để tạo một file Project mới.



**Hình PLA.6. Giao diện tạo Project help (tiếp).**

## Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop

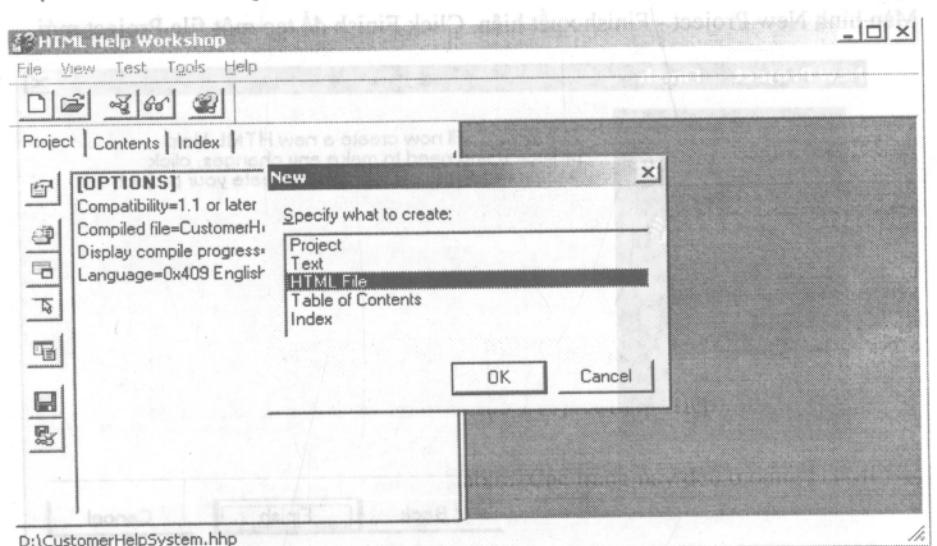
7. Màn hình sau sẽ xuất hiện. Tab Project được kích hoạt theo ngầm định.



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.

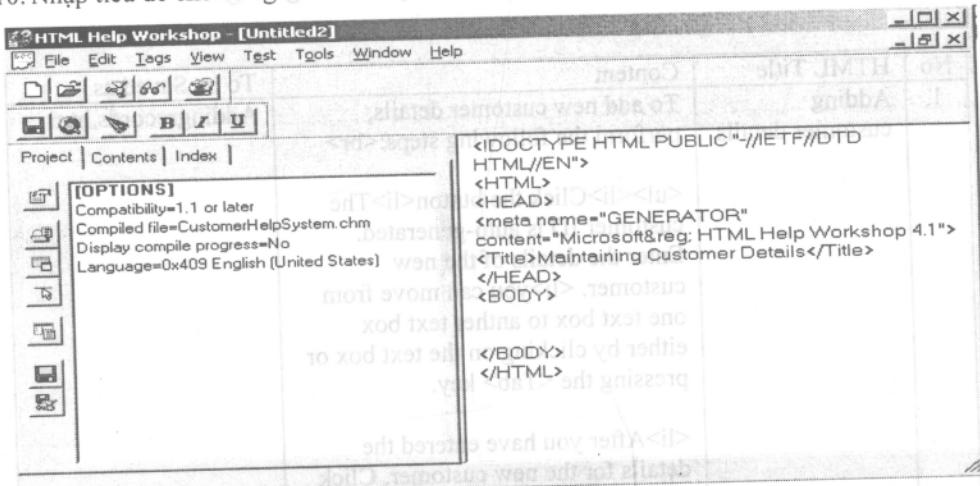
8. Để tạo một trang HTML, click File → New
9. Chọn HTML file từ hộp thoại New, click OK



Hình PLA.8. Giao diện tạo Project help (tiếp).

## Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop

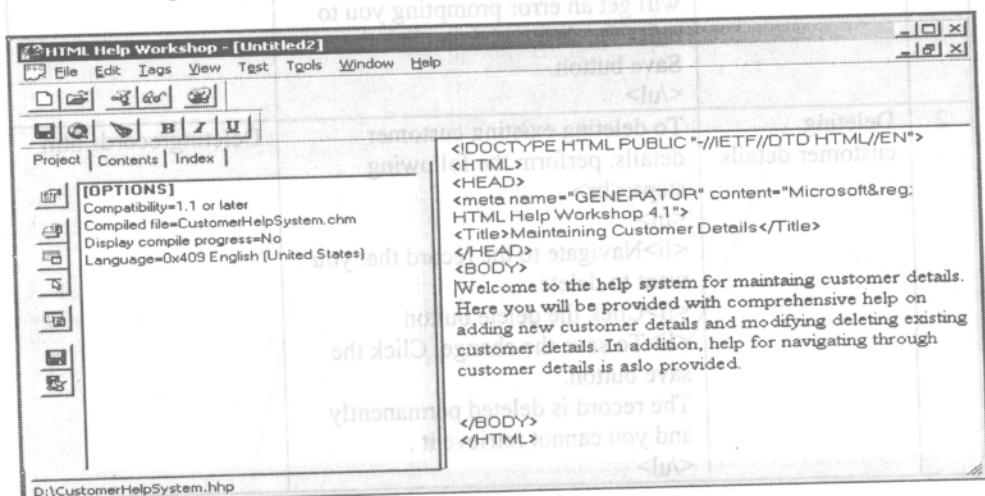
10. Nhập tiêu đề cho trang HTML là Maintaining customer details và click OK.



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 maintaining 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 also 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:&lt;br&gt;</p> <p>&lt;ul&gt;&lt;li&gt;Click the button&lt;/li&gt;The customer ID is auto-generated. Enter the details of the new customer. &lt;li&gt;you can move from one text box to another text box either by clicking on the text box or pressing the &lt;Tab&gt; key.</p> <p>&lt;li&gt;After you have entered the details for the new customer, Click the save button to save the entered details. &lt;br&gt;</p> <p>You cannot leave the first name and the address of the customer blank. &lt;br&gt;</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>	Addingrecords.htm
2	Deleting customer details	<p>To deleting existing customer details, perform the following steps:&lt;br&gt;</p> <p>&lt;ul&gt;</p> <p>&lt;li&gt;Navigate to the record that you want to delete.</p> <p>&lt;li&gt;Click the delete button</p> <p>&lt;li&gt;To save the change, Click the save button.</p> <p>The record is deleted permanently and you cannot retrieve it .</p>	Deletingrecords.htm

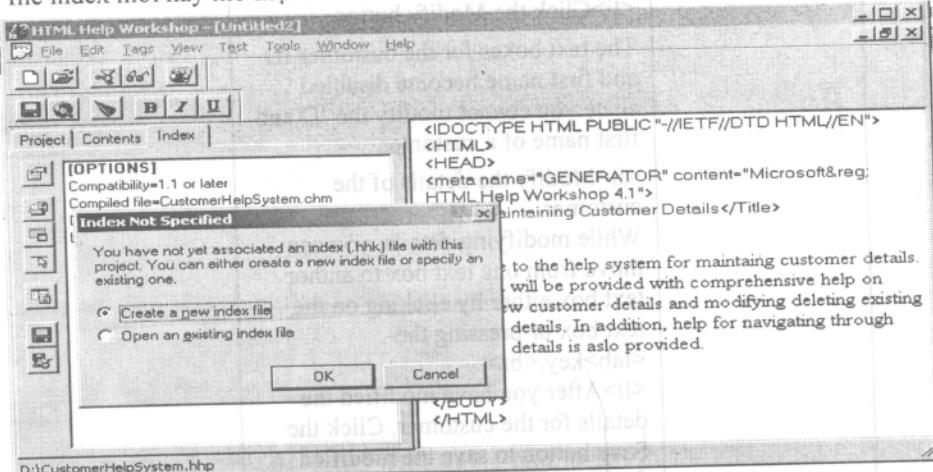
**Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop**

3	Modifying Customer details	<p>To modify existing customer details, perform the following steps:&lt;br&gt;&lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Navigate to the record that you want to modify</li> <li>&lt;li&gt;Click the Modify button.</li> </ul> <p>The text boxes for the customer ID and first name become disabled since you cannot modify the ID and first name of a customer.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Modify the details of the customer.</li> </ul> <p>While modifying details, you can move from one text box to another text box either by clicking on the text box or pressing the &lt;tab&gt;key.&lt;br&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;After you have modified the details for the customer, Click the Save button to save the modified details.&lt;br&gt;</li> </ul> <p>You cannot leave the first name and the address of the customer blank.</p> <p>&lt;br&gt;</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>	Modifyingrecords.htm
4	Navigating through customer details	<p>Use the following buttons to navigate through customer details:&lt;br&gt;&lt;br&gt;&lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;First – Use this button to view the first customer record. &lt;br&gt;&lt;br&gt;</li> <li>&lt;li&gt;Next – Use this button to view the record next to the record that you are currently viewing.&lt;br&gt;&lt;br&gt;</li> <li>&lt;li&gt;Previous – Use this button to view the previous customer record.&lt;br&gt;&lt;br&gt;</li> <li>&lt;li&gt;Last – Use this button to view the last customer record.&lt;br&gt;&lt;/ul&gt;</li> </ul>	Navigatingrecords.htm

## Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop

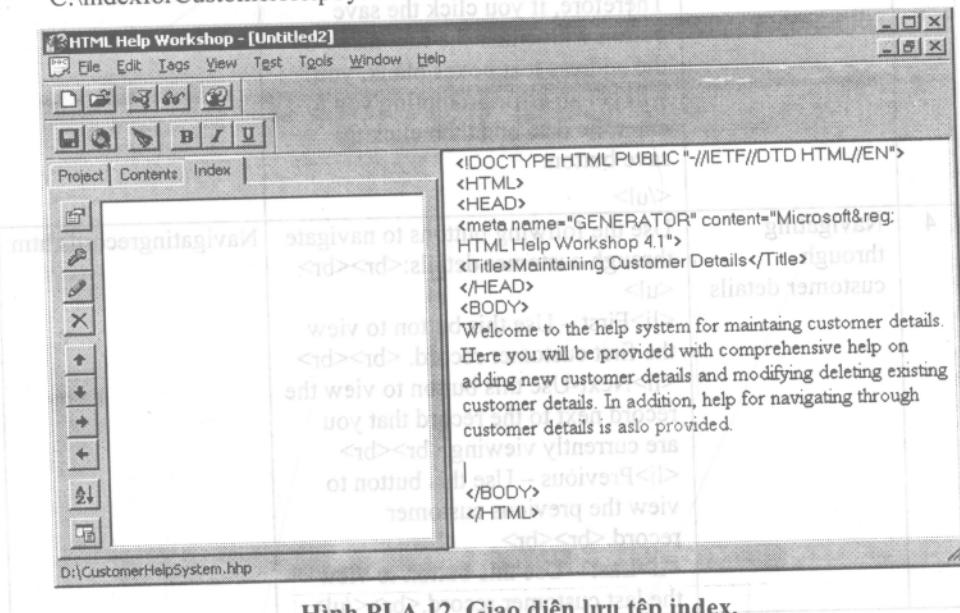
Sau khi các file HTML được tạo ra, bạn phải thêm vào các tính năng giúp người dùng xem một trang HTML bằng cách vào một từ khoá. Vì vậy, một file index cần được tạo ra cho các trang HTML.

13. 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.

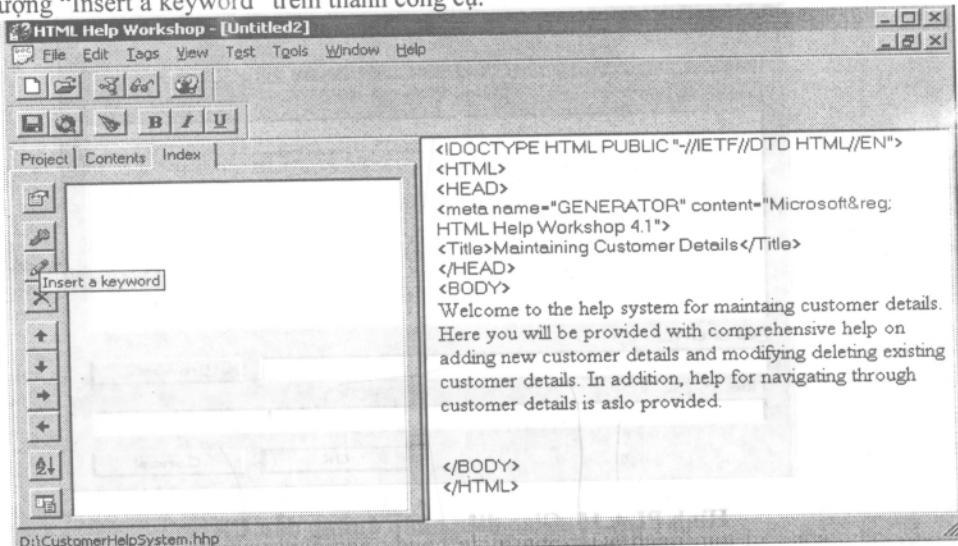
14. Mặc định, chọn “Create a new index file”. Click OK.  
15. Trong hộp thoại Save, nhập tên và đường dẫn cho file index  
“C:\indexforCustomerHelpSystem” và click Save.



Hình PLA.12. Giao diện lưu tệp index.

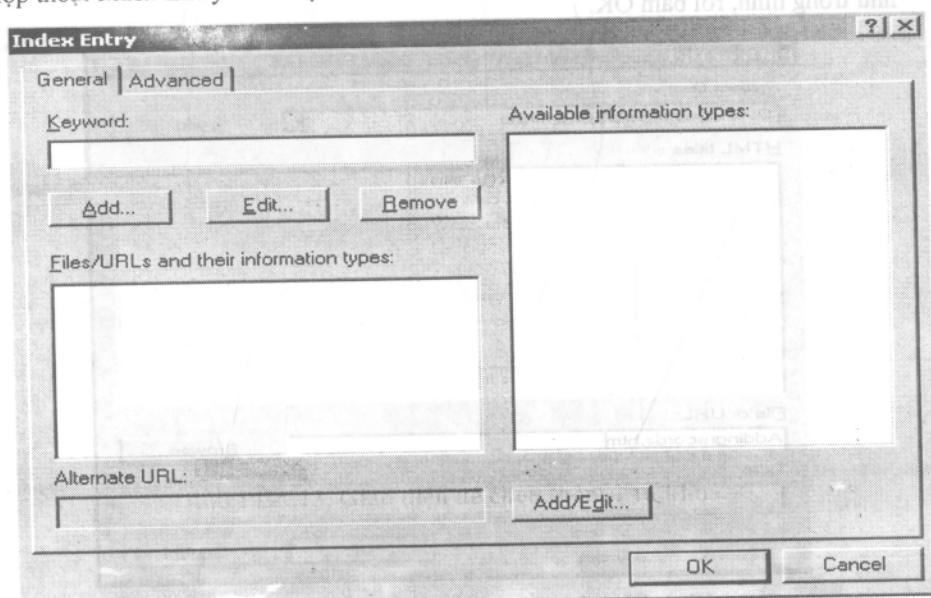
## *Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop*

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.



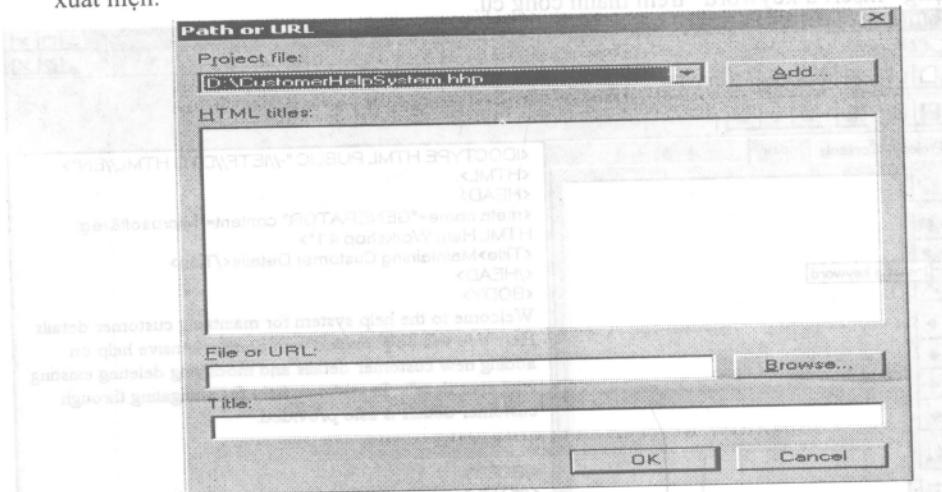
**Hình PLA.13. Giao diện để chèn thêm 1 từ khoá.**

17. Hộp thoại Index Entry xuất hiện



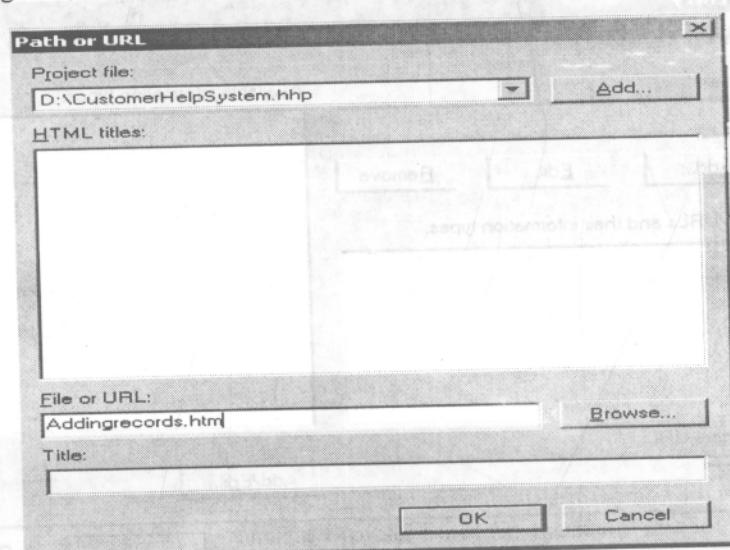
**Hình PLA.14. Giao diện để chèn thêm 1 từ khoá (tiếp).**

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.**

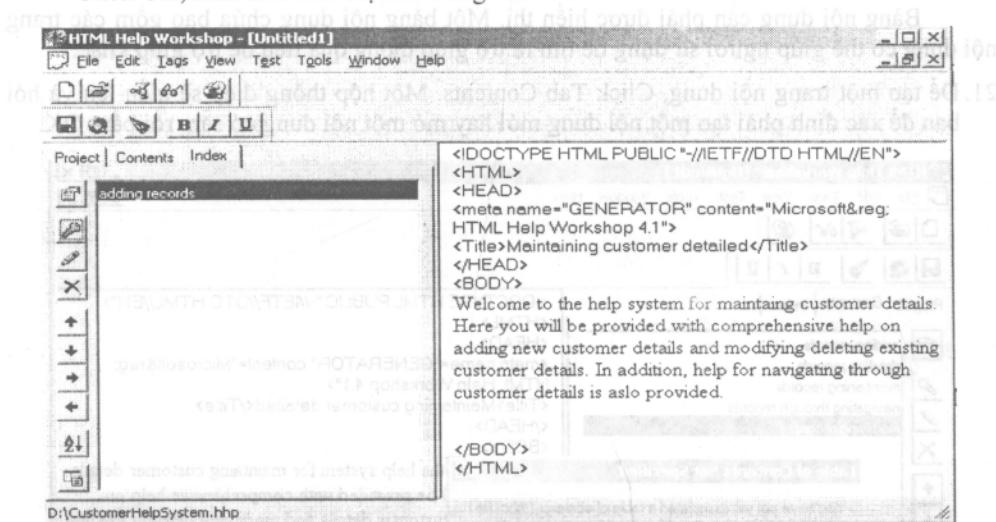
19. Bảng mặc định, Project đang mở được hiển thị trong list box của file project. Sau đó nhập tên file HTML kết hợp với từ khoá bởi enter tên file trong text File or URL hoặc là chọn lấy file bằng cách bấm nút Browse. Enter tên file name là addingrecords.htm như trong hình, rồi bấm OK.



**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.

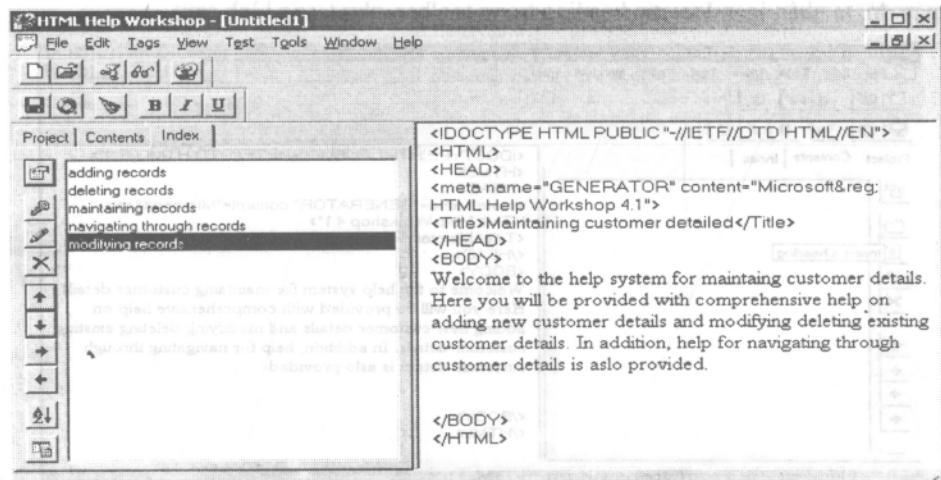


**Hình PLA.17. Giao diện toàn trang HTML và tệp lưu trữ.**

20. Tương tự, ta chèn vào những từ khóa sau:

Keyword	File to linked with the keyword
Deleting record	Deletingrecords.htm
Maintaining records	Maintainingrecords.htm
Navigating through records	Navigatingrecords.htm
Modifying records	Modifyingrecords.htm

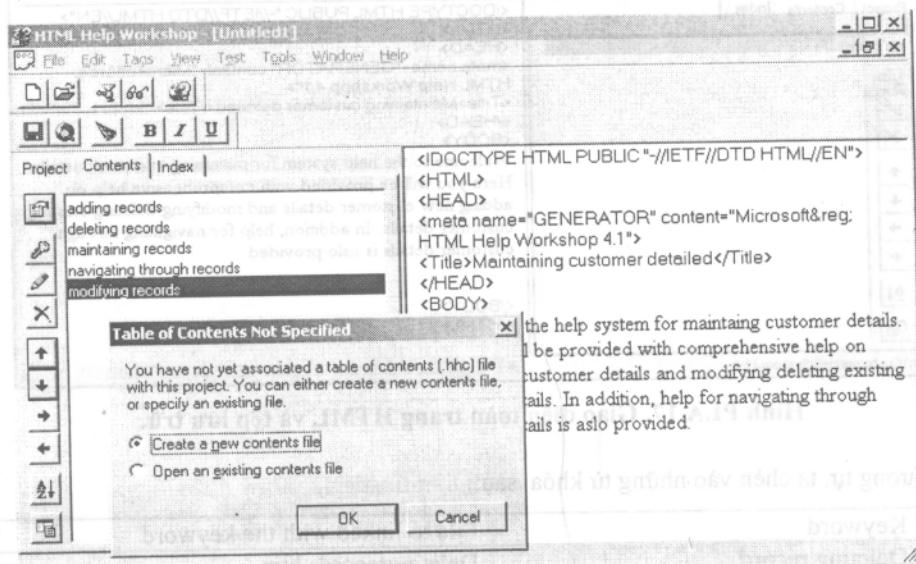
Màn hình có dạng:



**Hình PLA.18. Giao diện toàn trang HTML sau khi bổ sung các từ khoá.**

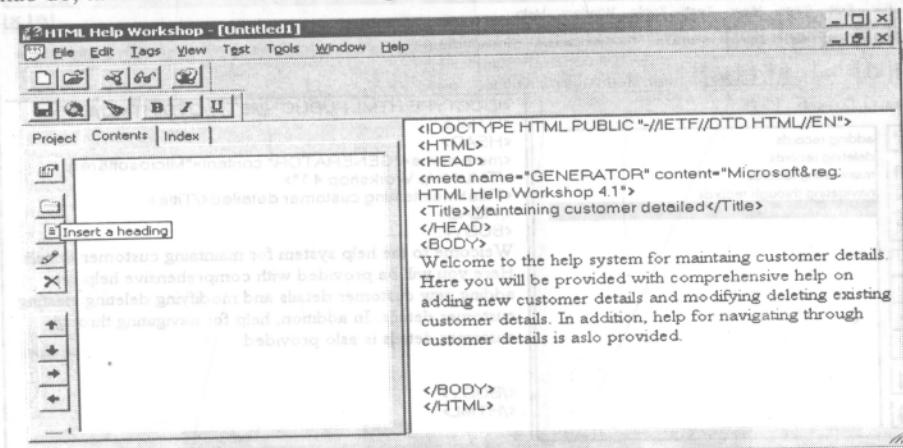
Bảng nội dung cần phải được hiển thị. Một bảng nội dung chứa bao gồm các trang nội dung có thể giúp người sử dụng để tìm ra trợ giúp thông qua tiêu đề trợ giúp khác.

21. 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.**

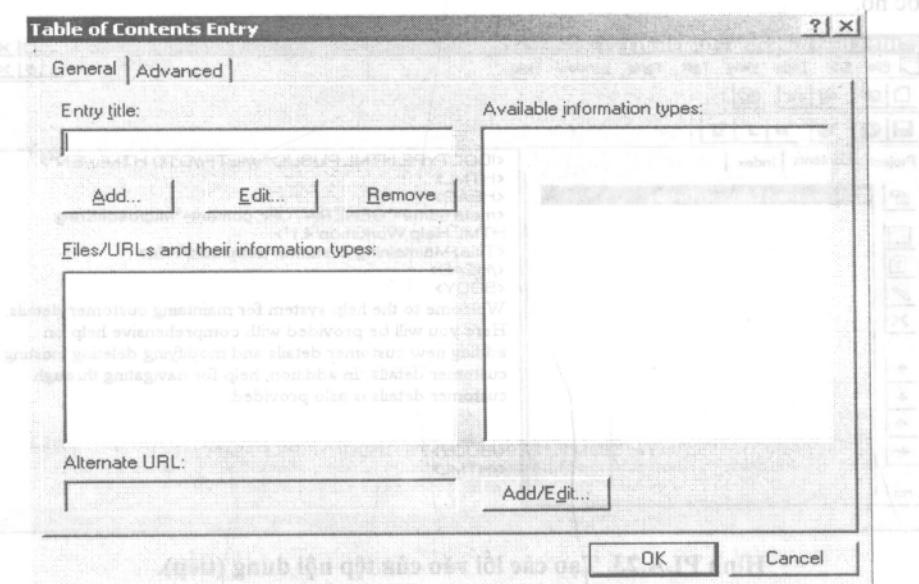
22. Nhập đường dẫn và tên của bảng file nội dung là D:\tocforCustomerHelpSystem và lưu lại.  
 23. Để tạo một heading của file nội dung phục vụ cho việc tìm kiếm của người sử dụng nào đó, ta nhấn icon Insert a heading trong toolbar, như trong hình sau:



**Hình PLA.20. Giao diện chèn tiêu đề trang.**

## **Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop**

24. Hộp thoại Table Of Contents Entry sẽ xuất hiện như trong hình PLA.19.

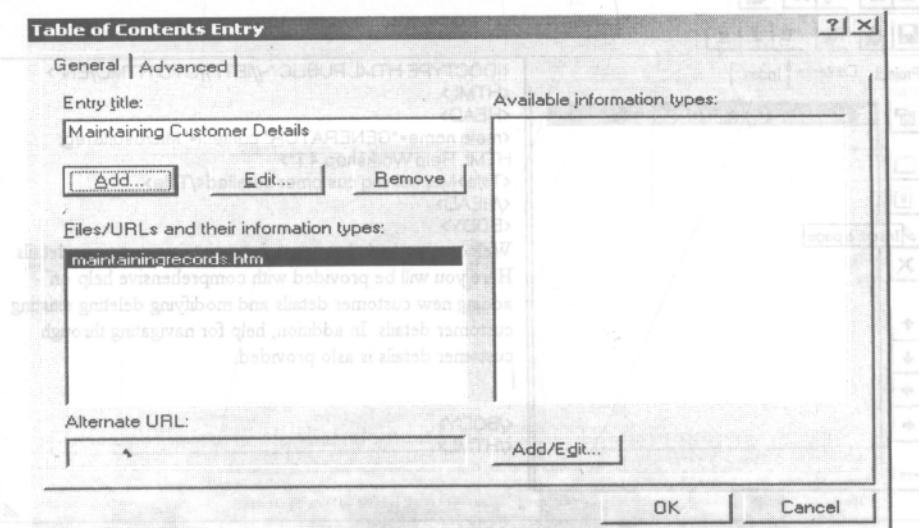


**Hình PLA.21. Tạo các lối vào của tệp nội dung.**

25. Nhập từ "Maintaining 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.

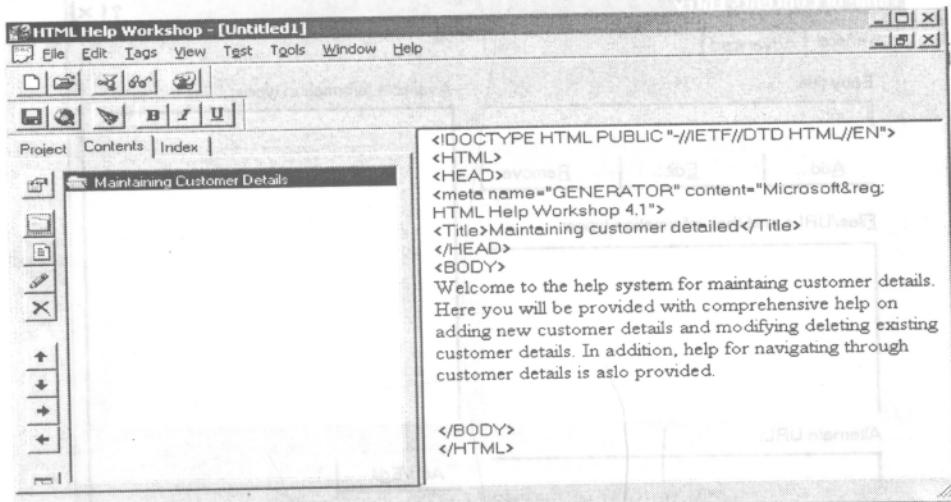
27. Hộp thoại Table Of Contents Entry sẽ hiển thị như hình PLA.20.



**Hình PLA.22. Tạo các lối vào của tệp nội dung (tiếp).**

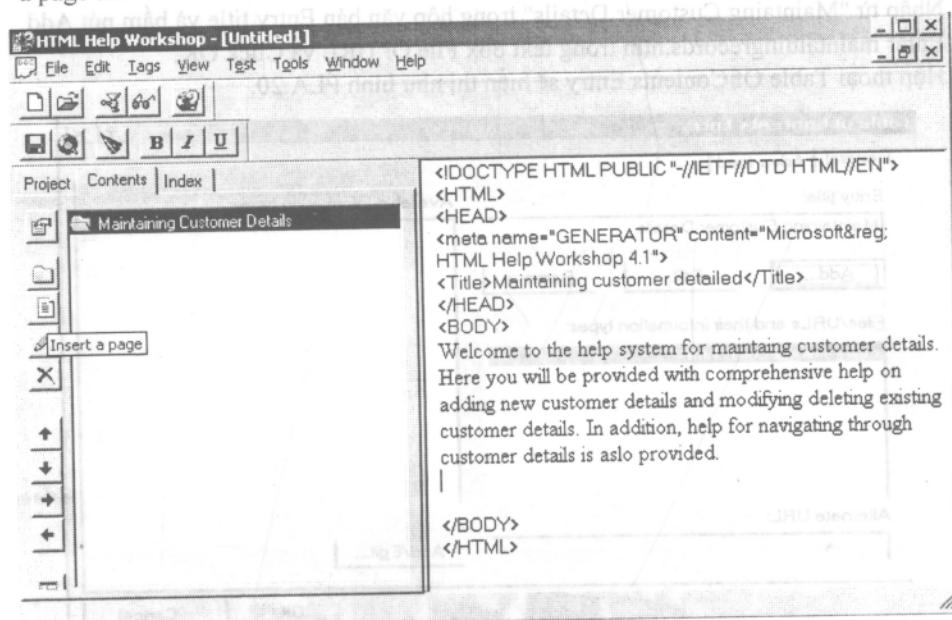
## Phụ lục A: Tạo hệ trợ giúp HTML Help Workshop

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 lối vào của tệp nội dung (tiếp).

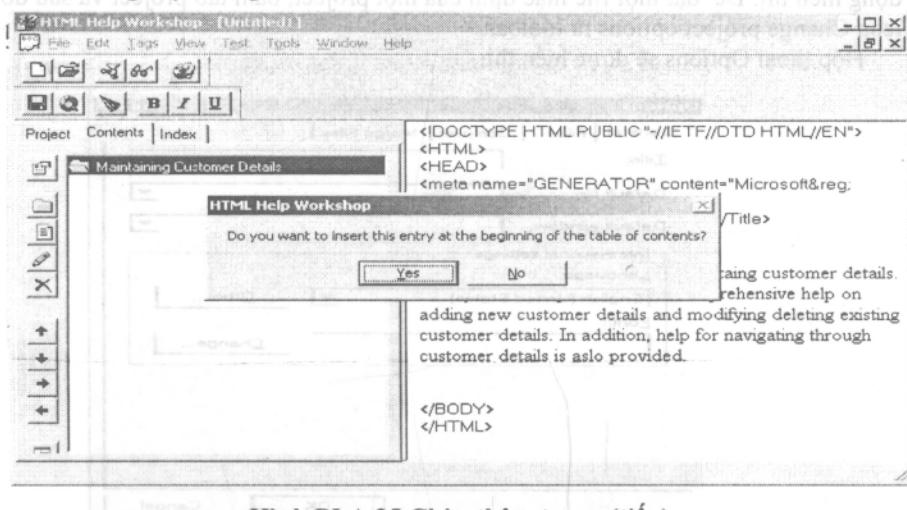
28. Sau khi heading được tạo, nội dung của các trang phải được nhập vào. Nhấn icon Insert a page trên toolbar.



(quay) 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?



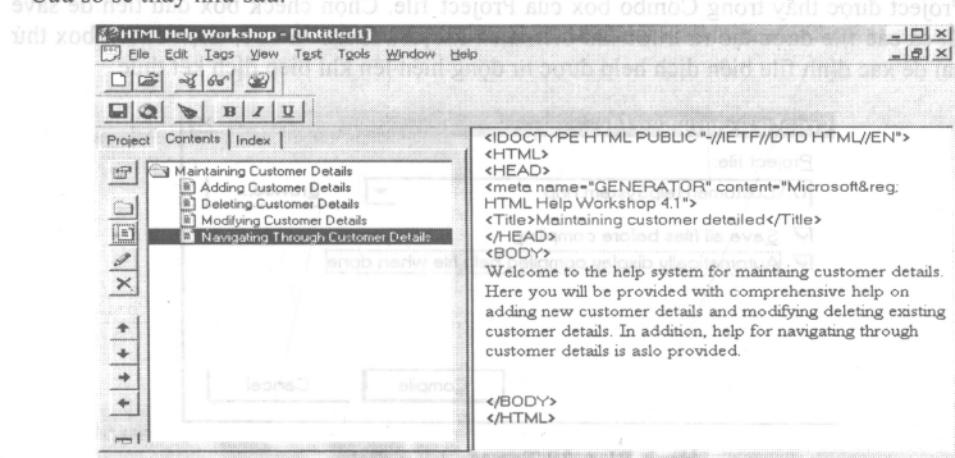
Hình PLA.25.Chèn thêm trang (tiếp).

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
Adding Customer Details	Addingrecords.htm
Modifying Customer Details	Modifyingrecords.htm
Navigating Through Customer Details	Navigatingrecords.htm

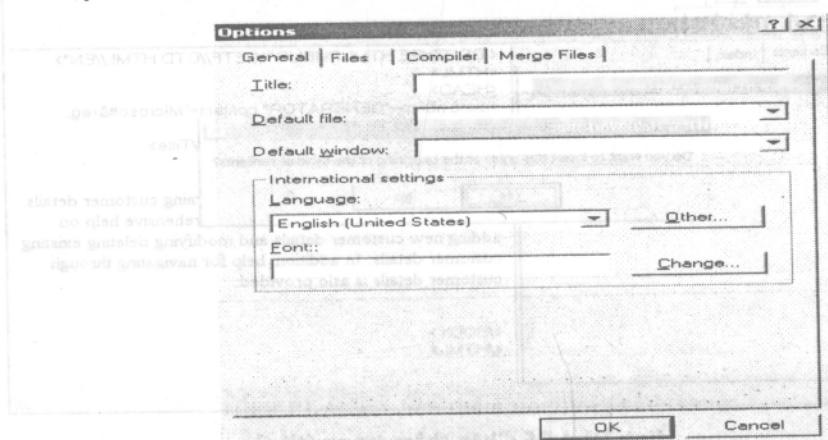
Cửa sổ sẽ thấy như sau:



Hình PLA.26.Chèn thêm nội dung vào trang.

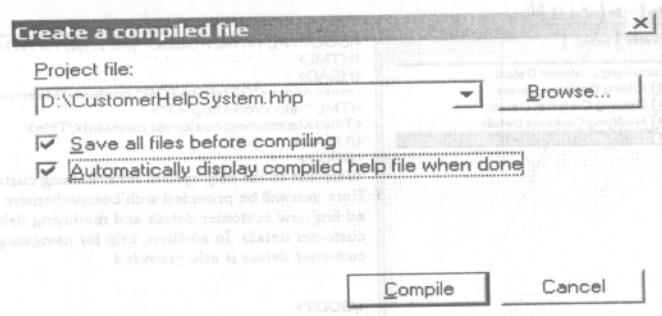
32. Bạn có thể chỉ định một file HTML mặc định của project bạn. Khi file help được hiển thị của một người sử dụng nào đó, file HTML đặt mặc định của một Project được mở và tự động hiển thị. Để đặt một file mặc định của một project, bấm tab project và sau đó bấm icon Change project options từ toolbar.

Hộp thoại Options sẽ được hiển thị:



**Hình PLA.27.Tạo hộp hội thoại.**

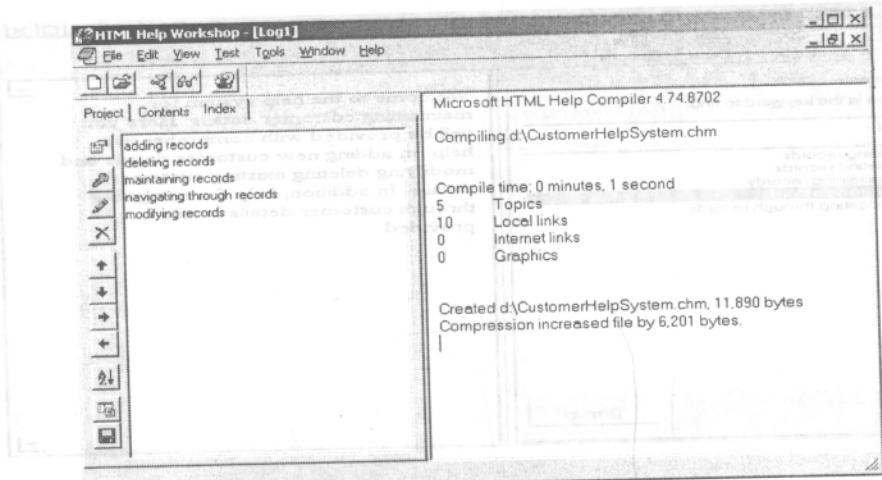
33. Enter file maintainingrecords.htm là file mặc định trong Combo box Default File và bấm OK.
34. Sau khi file index và bảng nội dung được thêm vào trong Project, bạn cần dịch Project file để insert và save tất cả các files trong folder Project. Để biên dịch Project, bấm vào File từ thanh menu và chọn 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:

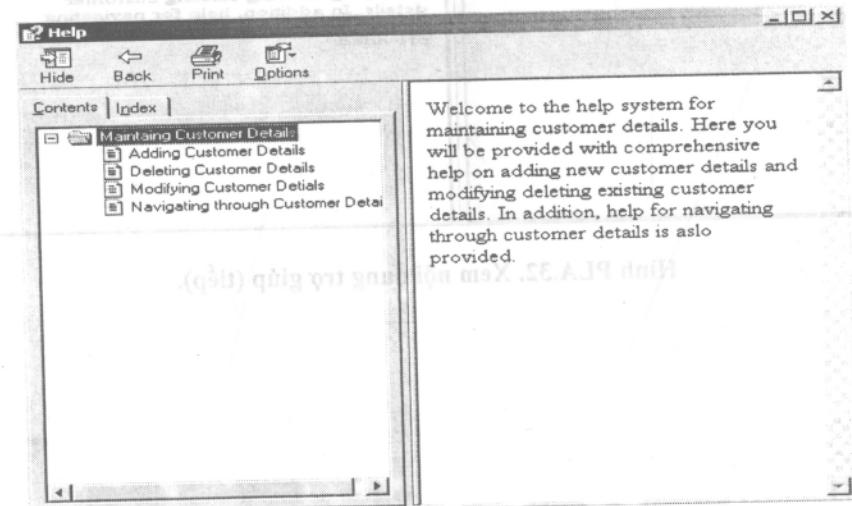


Hình PLA.29. Cửa sổ đe án đã biên dịch.

### 4. Truy nhập hệ thống trợ giúp

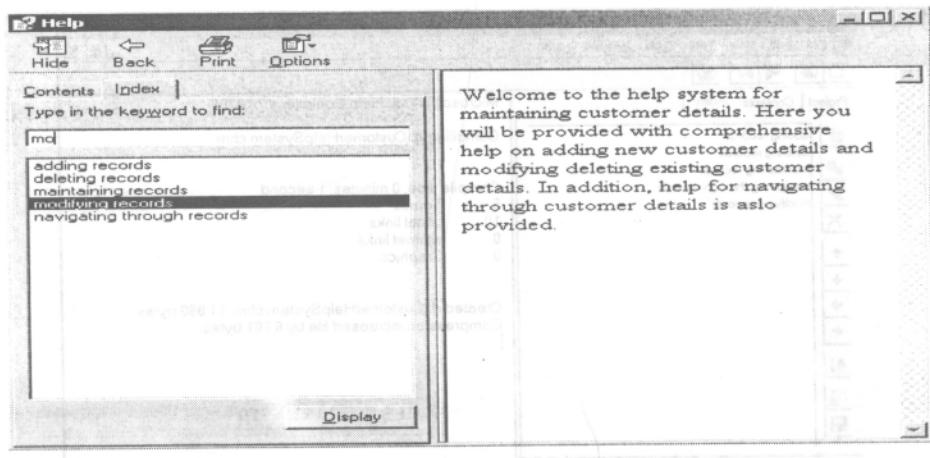
#### Hành động:

- Để xem file help đã biên dịch, bạn click vào View từ thanh menu. Sau đó chọn Compiled File từ menu View. Enter đường dẫn của file help trong Combo box của Compiled File.
- Sau khi bấm View rồi nó sẽ hiển ra như hình sau:



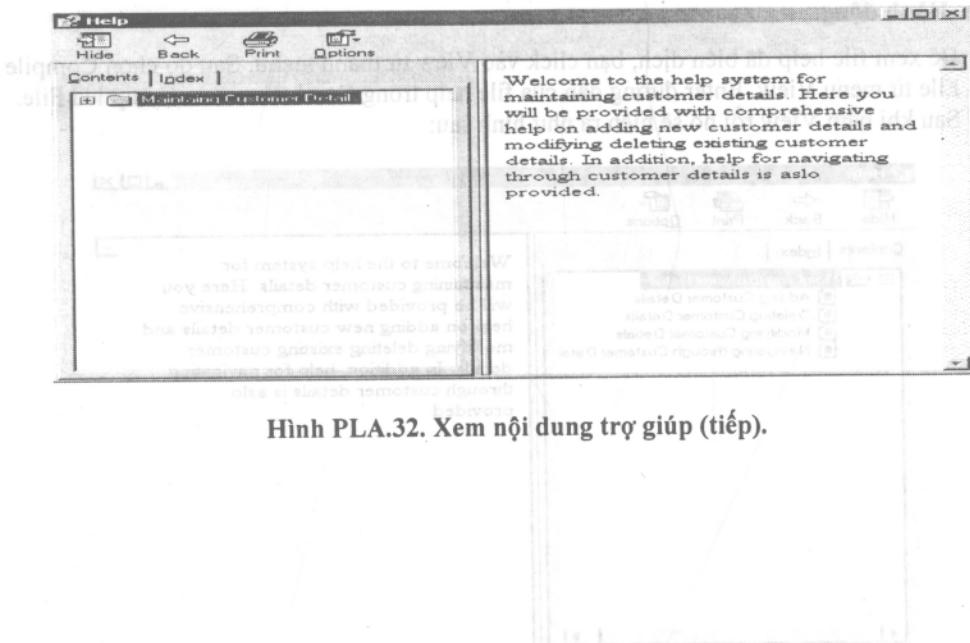
Hình PLA.30. Nội dung tệp trợ giúp đã biên dịch.

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:

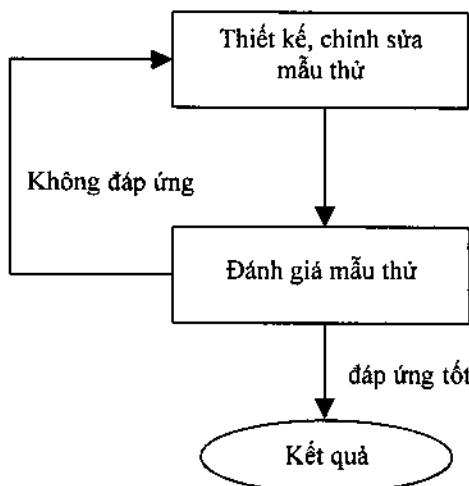


**Hình PLA.32. Xem nội dung trợ giúp (tiếp).**

## **Phụ lục B**

### **CÁC CÔNG CỤ TẠO LẬP MẪU THỬ HYPERCARD VÀ REVOLUTION**

Trong thiết kế giao diện, việc sử dụng mẫu thử là cần thiết. Ở đó, người ta đưa ra các ý kiến của mình thông qua việc trình diễn của các mẫu thử, các yêu cầu người dùng về giao diện qua đó được xác định lại. Quy trình xây dựng mẫu thử được chỉ ra như trong hình PLB.1.



**Hình PLB.1. Mô hình mẫu thử giao diện**

Có khá nhiều công cụ xây dựng mẫu thử, 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.

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 (đề 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 điều khiển 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 điện 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 hiện 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

## *Phụ lục B: Công cụ tạo lập mẫu thử*

---

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 mouseClick
    go next
    wait for 100 millisecond
end repeat
```

3. Lệnh switch

## Phụ lục B: Công cụ tạo lập mẫu thử

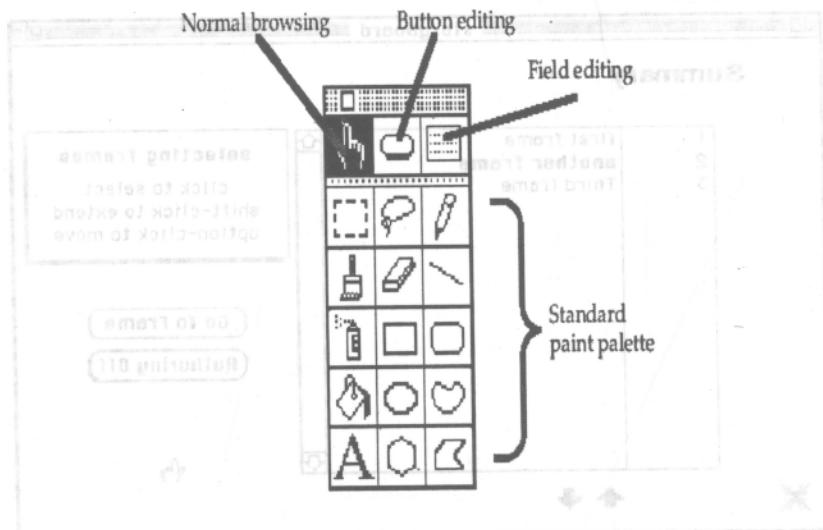
Cú pháp:

```
switch <expression>
    case <one>
        --các lệnh 1
    break
    case <two>
        --Các lệnh 2
    break
end switch
```

Ví dụ:

```
on menuPick pWhich
    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

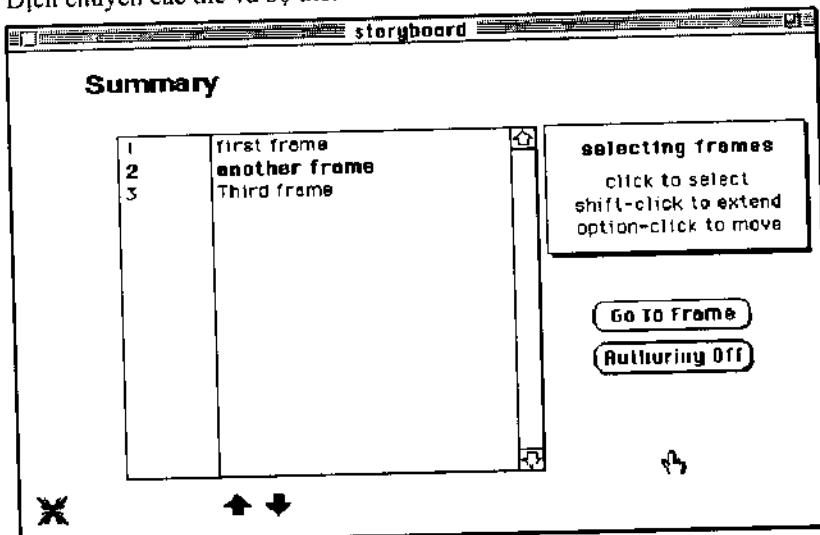
- Tên màn hình.
- Số trình tự.

Bảng điều khiển

- Di chuyển các thẻ.
- Âm 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.

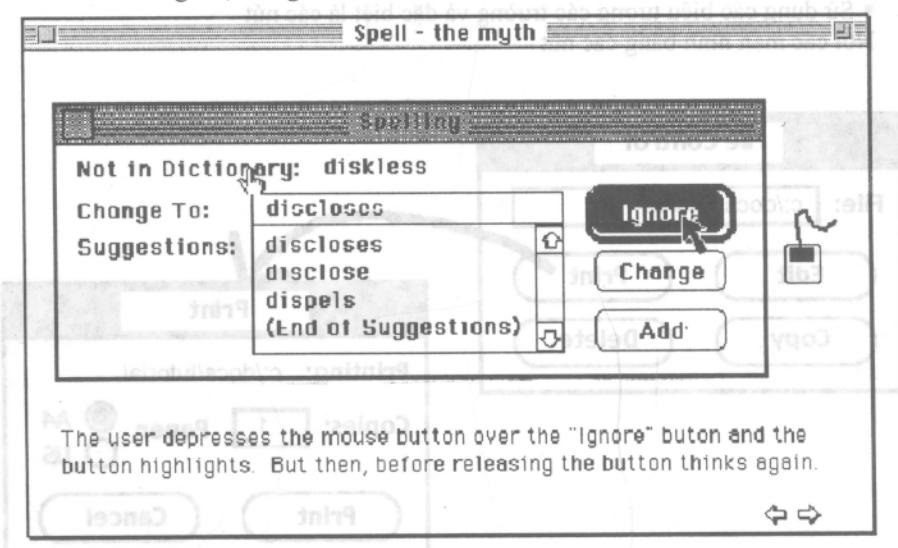


Hình PLB.3. Một storyboard đơn giản.

## 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

- Tất cả các trạng thái.
- Tất cả các đường dẫn.
- Tất 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

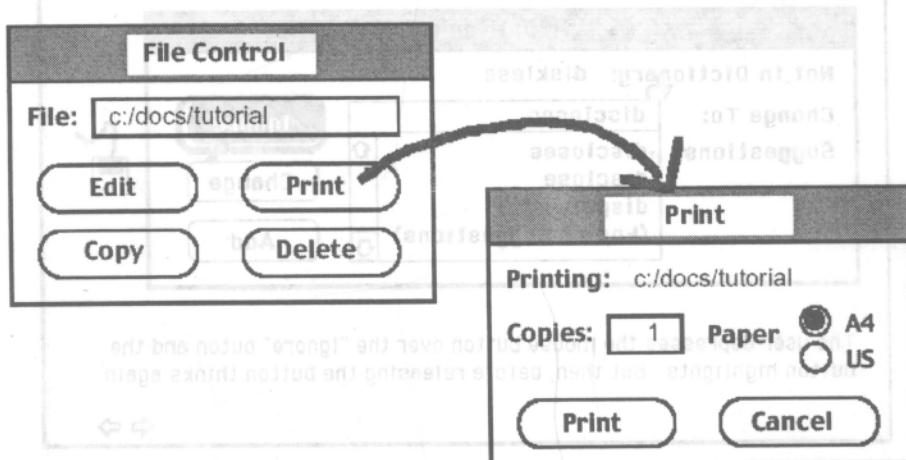
- Phác thảo các trạng thái chính.

## Phụ lục B: Công cụ tạo lập mẫu thử

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

Nối các màn hình bằng các nút

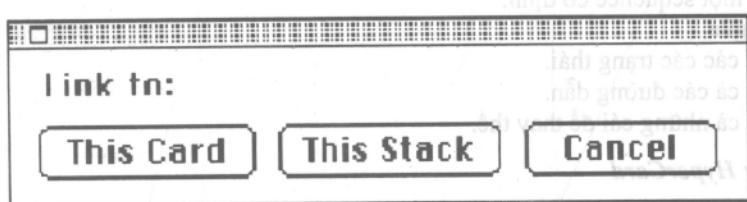


Hình PLB.5. Xây dựng giao diện điều khiển in.

### 3.10. Tạo link

- Chọn bộ các nút
- Chọn nút để nối
- Sử dụng 'Button info...' từ 'Objects'

Nhấn vào 'Link to...'

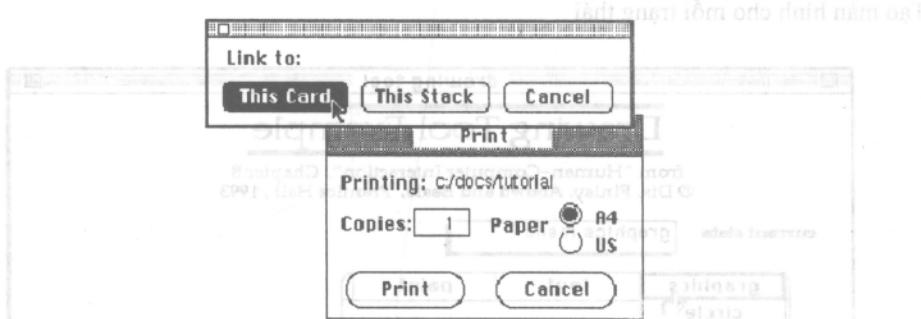


Hình PLB.6. Giao diện tạo Link.

+ Chuyển đến thẻ nhận

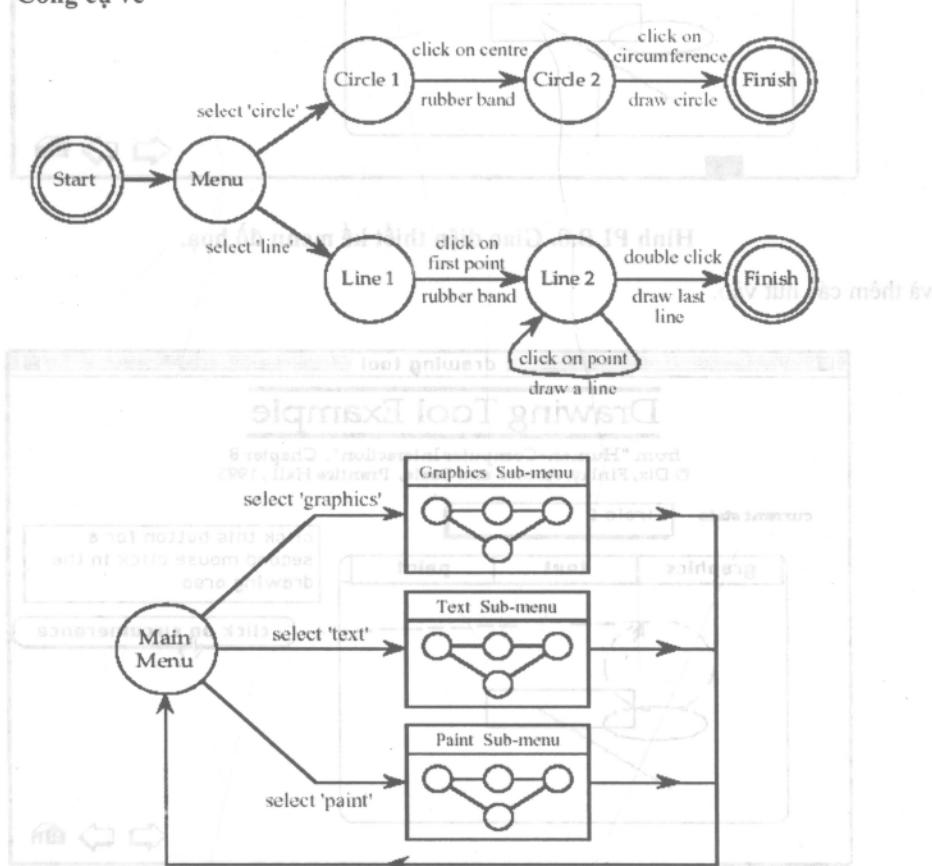
Ấn vào 'This card'

## Phụ lục B: Công cụ tạo lập mẫu thử



Hình PLB.7. Giao diện tạo Link (tiếp).

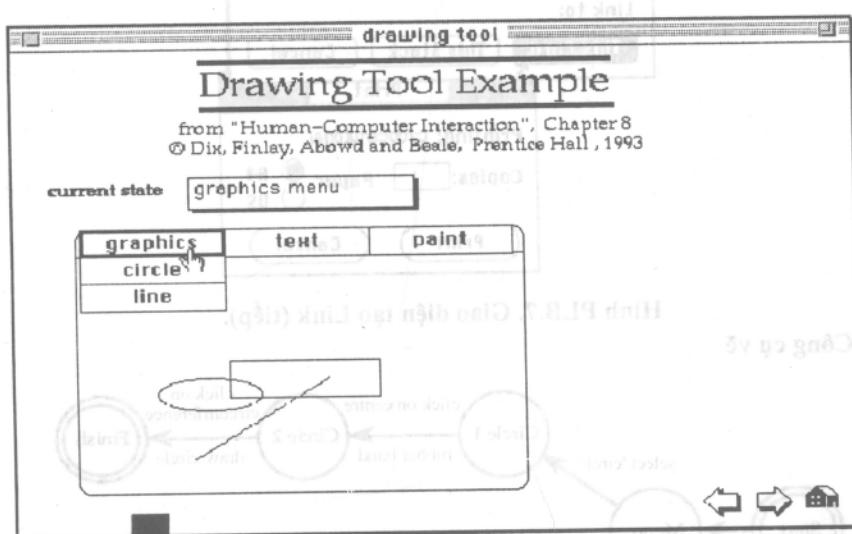
Ví dụ - Công cụ vẽ



Hình PLB.8. Các ví dụ tiếp theo về menu giao tiếp.

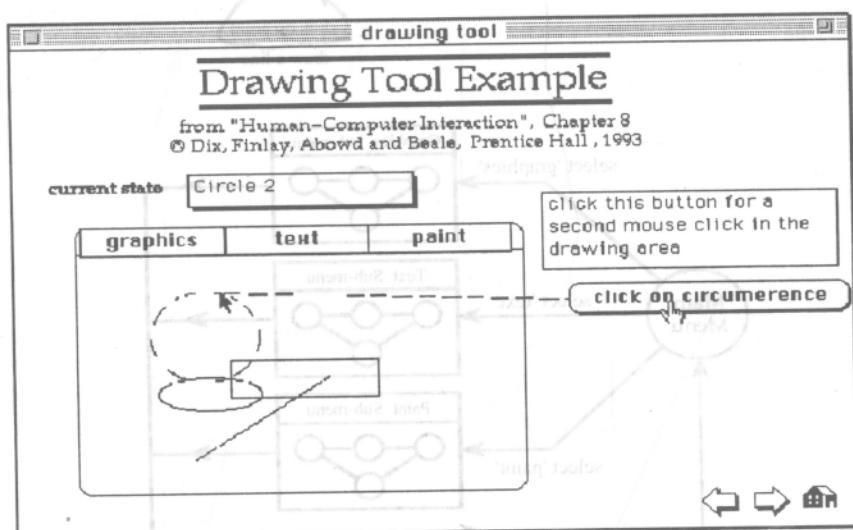
(from *Human-Computer Interaction*, Dix, Finlay, Abowd and Beale, 1993)

Tạo màn hình cho mỗi trạng thái



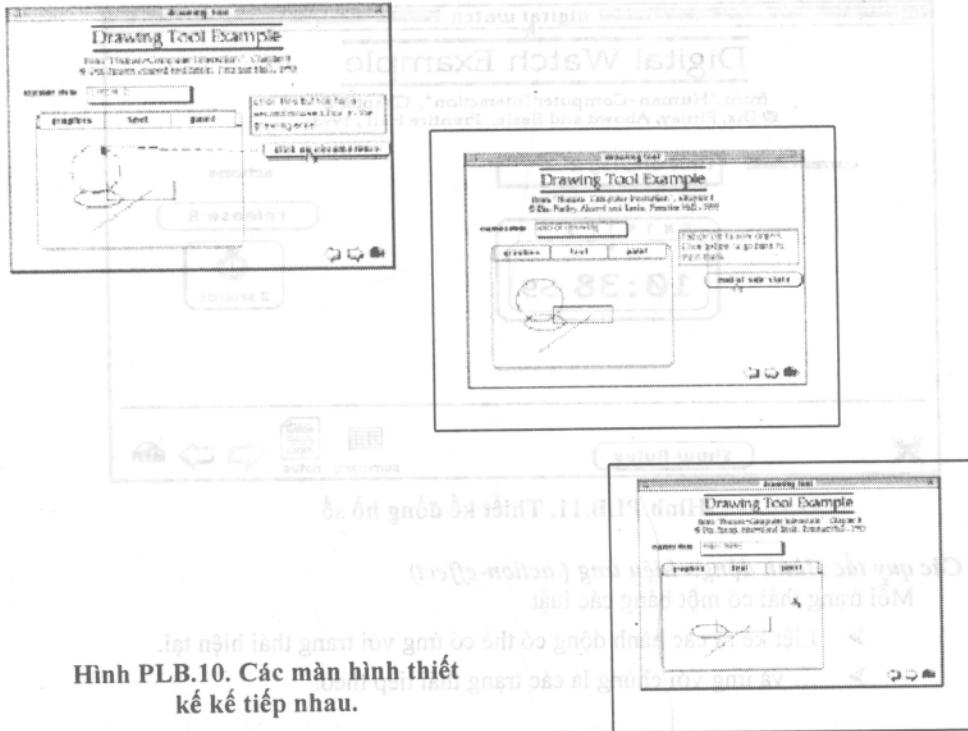
**Hình PLB.8. Giao diện thiết kế menu đồ họa.**

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

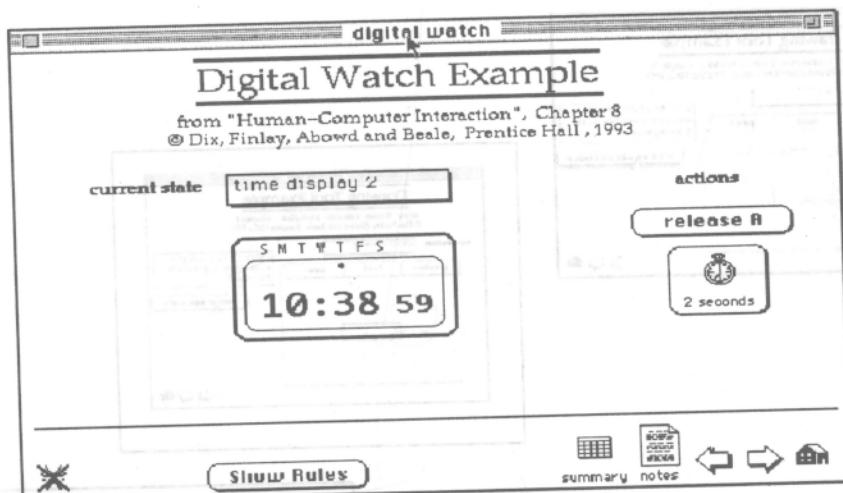
- Chỉ rõ dialogue.
- Thực hiện các chi tiết.
- Phân tích các chi tiết.

## 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.

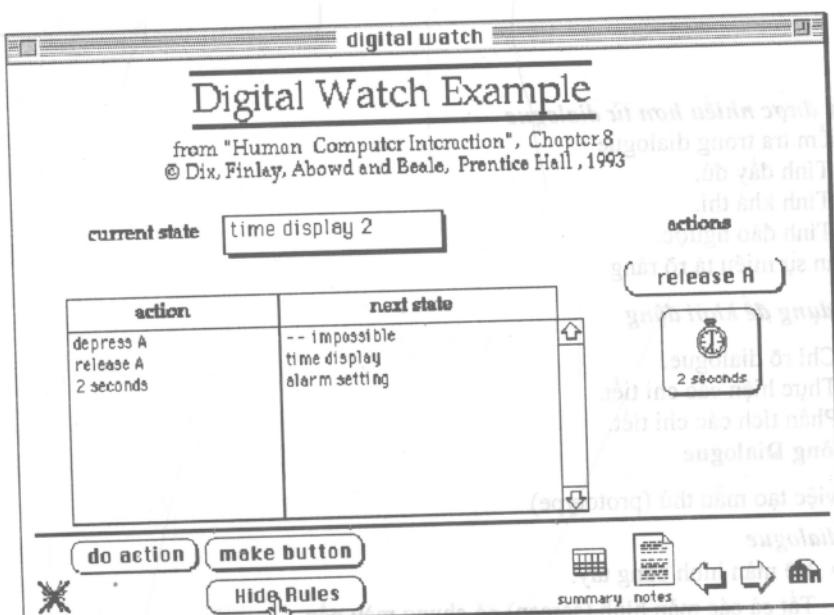


Hình PLB.11. Thiết kế đồng hồ số

#### 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.

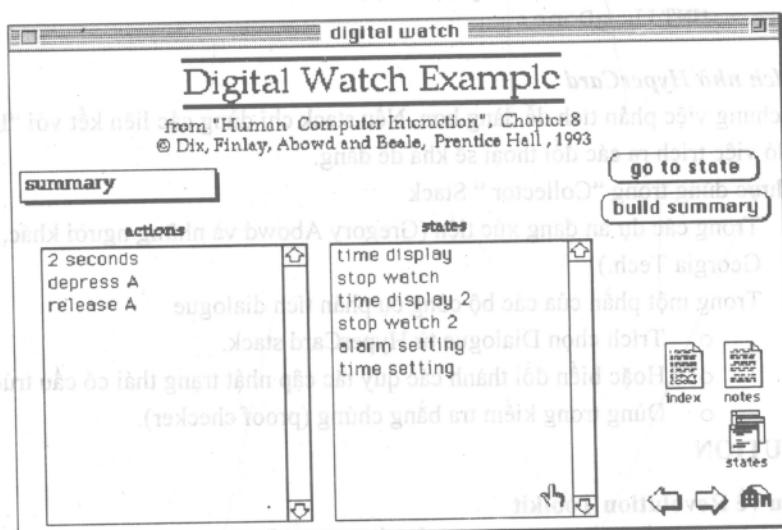
- Thay vì bảng ta có thể sử dụng.

- 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.message));
                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 thẻ 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 môi trường khá rộng: Window, Unix, Mac 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 chi 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 thẻ chứa các đối tượng: đồ họa, văn bản, phím lệnh, đa phương tiện và menu (hình PLB14).

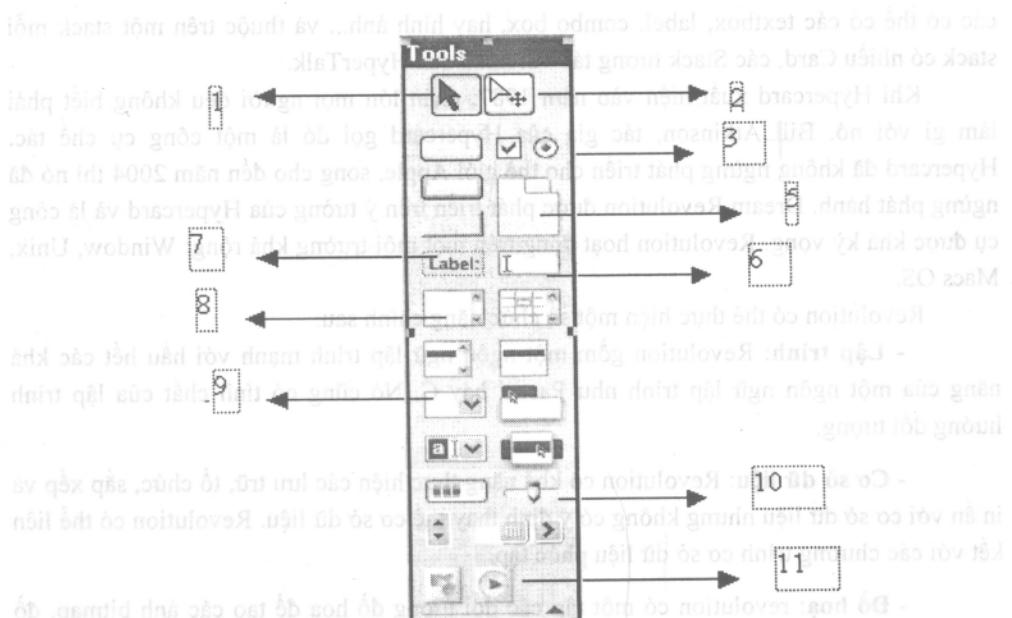
Đối tượng cơ bản khi xây dựng ứng dụng của revolution là Stack. Mọi cửa sổ nhìn thấy trong môi trường đều là Stack. Không có đối tượng nào tồn tại độc lập với Stack. Do vậy, bước đầu tiên để tạo một ứng dụng là tạo một stack (sử dụng Menu File- New Maintstack).

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.

## Phụ lục B: Công cụ tạo lập mẫu thử



**Hình PLB.14. Panel điều khiển của Revolution Toolkit.**

1: chế độ run;

2: chế độ design;

3: các điều khiển checkbox và radiobox;

4: các nút và các tab;

5: hộp textbox;

6: label;

7: list box;

8: điều khiển combo Box;

9: các thanh tiến trình, thanh trượt;

10: 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 cấu trúc sau:

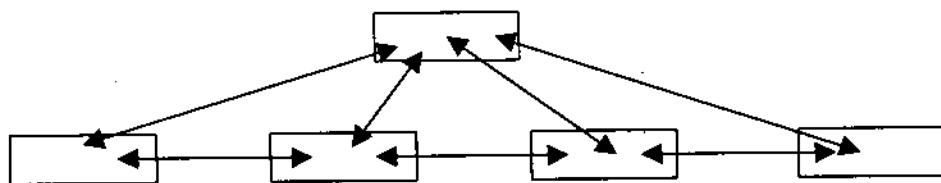
- Tuyến tính
- Nhảy tuyến tính (Jump linear)

- 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.

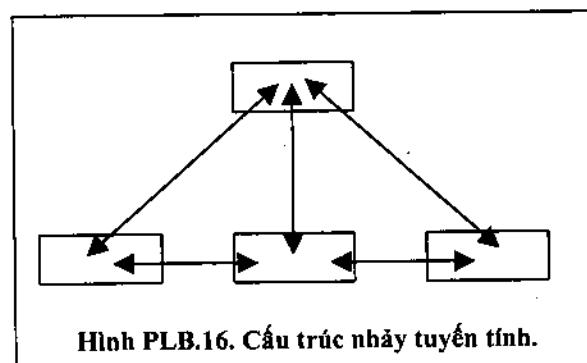


**Hình PLB.15. Cấu trúc tuyến tính.**

### 2.2. Cấu trúc nhảy tuyến tính

Cấu trúc này bao gồm:

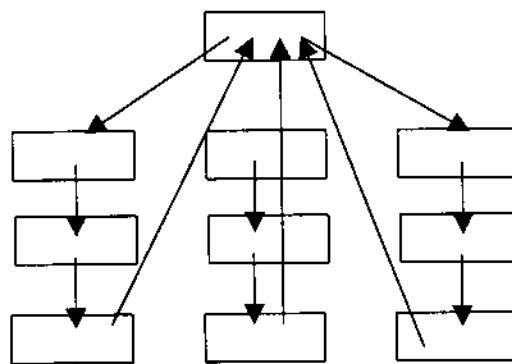
- mainstack;
- 3 substacks;
- một nhóm trong mỗi substack chứa tất cả các trường và phím lệnh;
- Có đủ số card cần thiết trên mỗi stack với nhóm trên mỗi stack.



### 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 phím lệnh;
- có đủ số card cần thiết trên mỗi substack với nhóm trên mỗi stack.

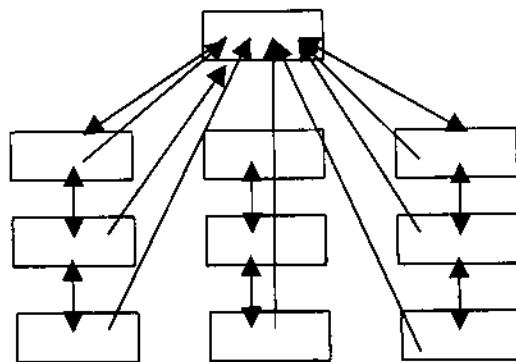


**Hình PLB.17. Cấu trúc nhánh chặt.**

### 2.4. Cấu trúc nhánh mở

Cấu trúc này bao gồm:

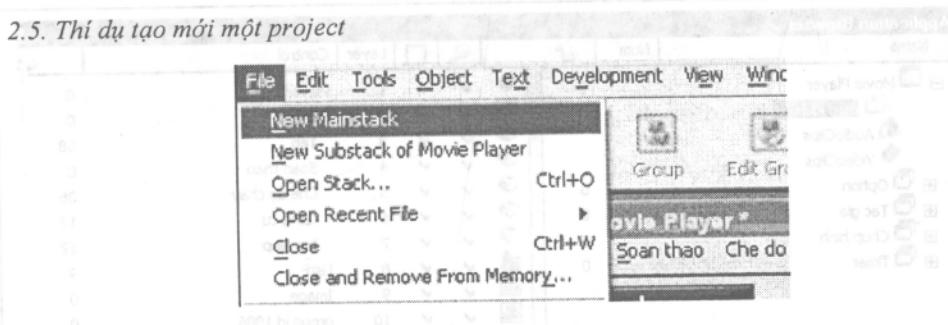
- mainstack;
- 3 substacks;
- một nhóm trong mỗi substack chứa tất cả các trường và phím lệnh;
- có đủ số card cần thiết trên mỗi substack với nhóm trên mỗi stack.



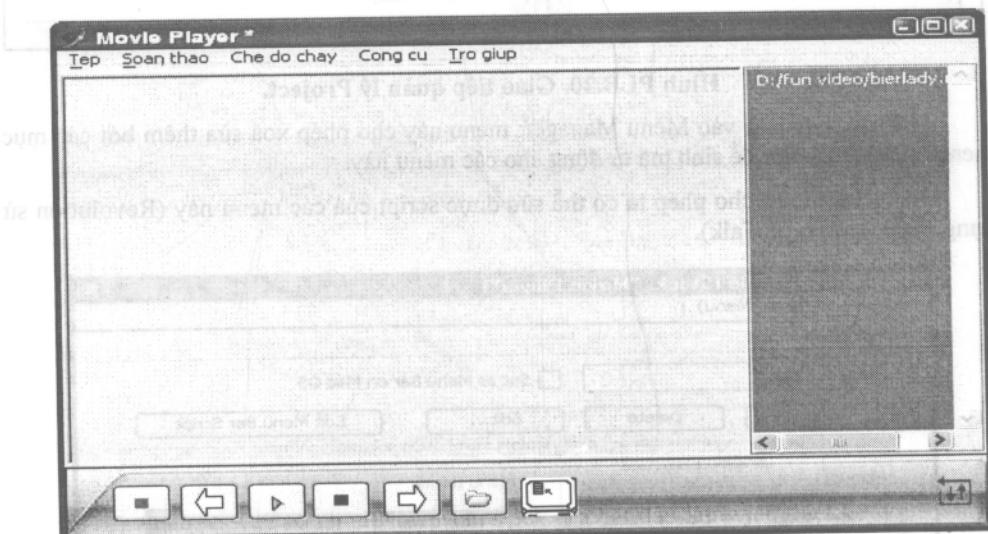
**Hình PLB.18. Cấu trúc nhánh mở.**

## Phụ lục B: Công cụ tạo lập mẫu thử

### 2.5. Thí dụ tạo mới một project

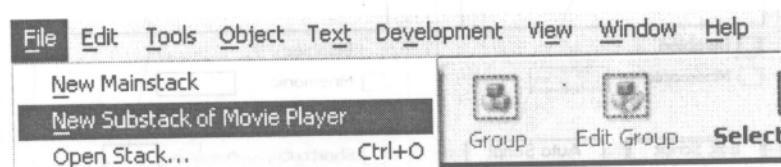


Vào cửa sổ file tạo chọn New Mainstack: Ta bắt đầu làm việc thiết kế với Mainstack này với giao diện rất trực quan và gần gũi, tạo được một mainstack như sau (hình PLB.19):



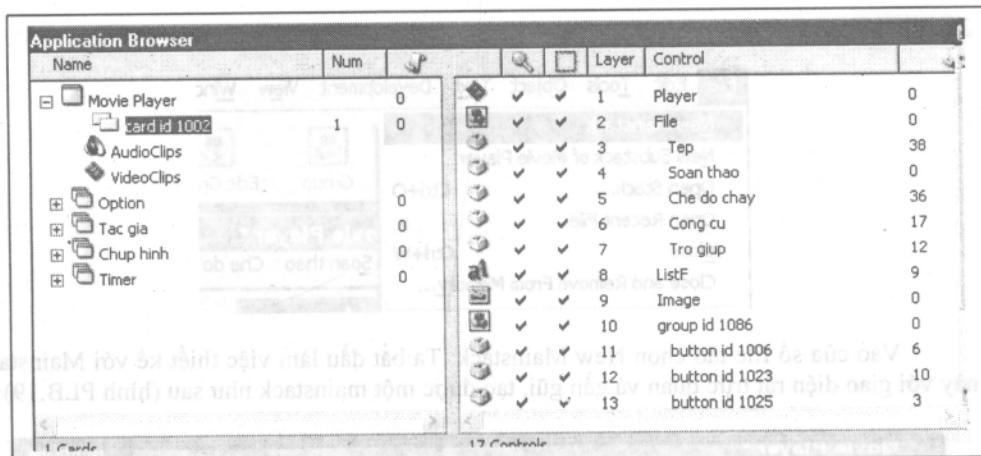
Hình PLB.19. Tạo Mainstack cho một ứng dụng Movie Player.

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 của sổ Application Browser. Ta dễ dàng có thể thấy rõ được cấu trúc thư mục của Project:

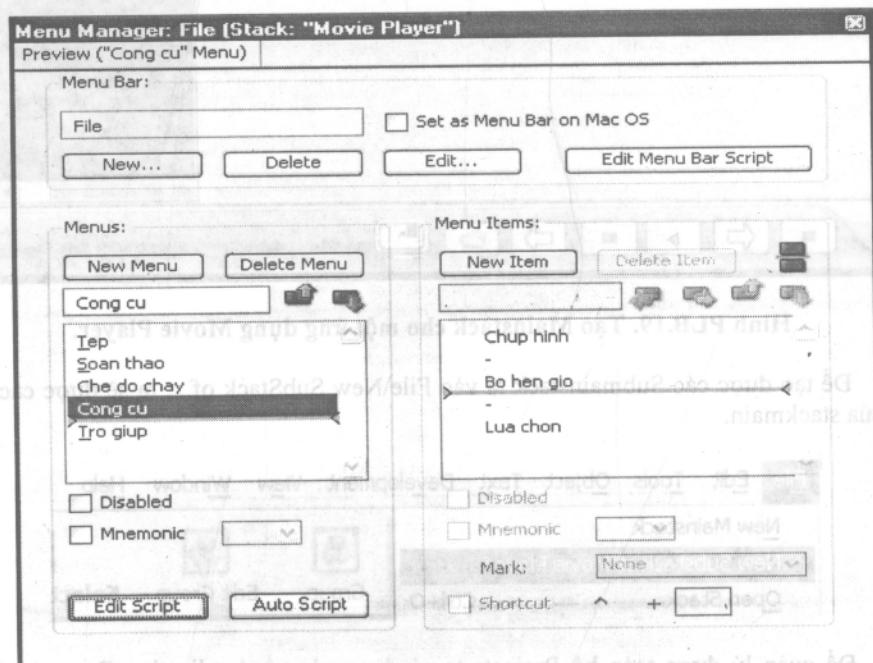
## Phụ lục B: Công cụ tạo lập mẫu thử



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.

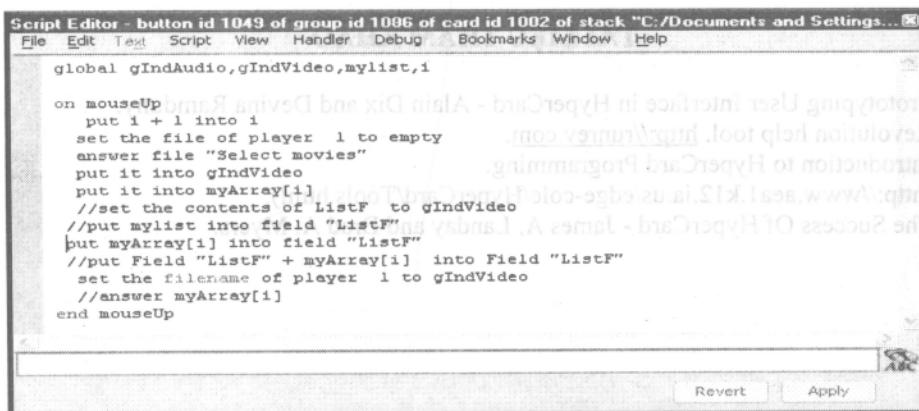
Nút EditScript cho phép ta có thể sửa được script của các menu này (Revolution sử dụng ngôn ngữ HyperTalk).



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

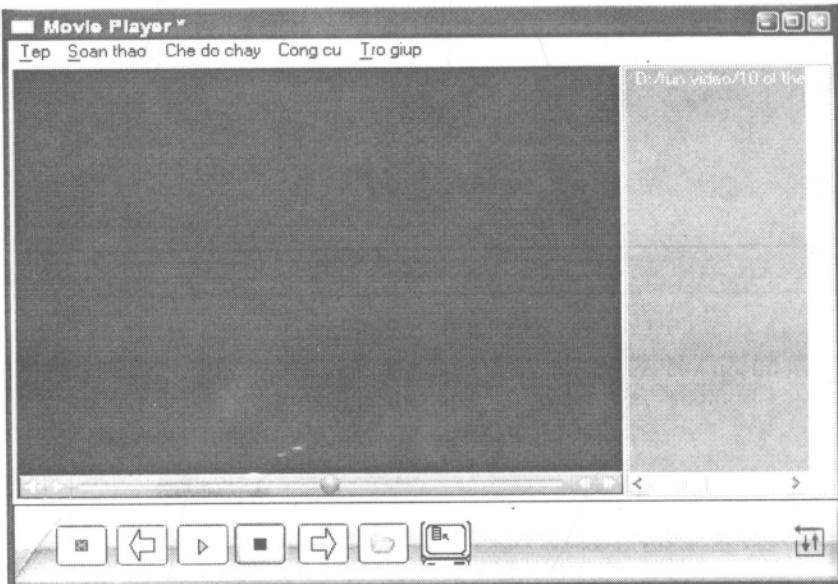
global gIndAudio,gIndVideo,mylist,i

on mouseUp
    put i + 1 into i
    set the file of player 1 to empty
    answer file "Select movies"
    put it into gIndVideo
    put it into myArray[i]
    //set the contents of ListF to gIndVideo
    //put mylist into field "ListF"
    put myArray[i] into field "ListF"
    //put Field "ListF" + myArray[i] into Field "ListF"
    set the filename of player 1 to gIndVideo
    //answer myArray[i]
end mouseUp
```

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 tuy 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://wwwaea1.k12.ia.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, Trưởng 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 thỏa 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 họa điện tử (electronic slide shows) và các prototype tương tác. Tiếp đó, kiểm tra tính sử tiện 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 - kiều 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 kiều phông không chân, khoảng cách giữa các ký tự nên tối thiểu là một lần nhấn 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 kiều 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 thi 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 nhắm 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 thạo 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.

- 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

<input checked="" type="checkbox"/> Analyse the opportunity	<input type="checkbox"/> Create the user experience
<input type="checkbox"/> Build the context of use	<input type="checkbox"/> Track usage and improve

### TYPE OF GUIDANCE

Specifications     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: 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 dụng 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 tiêu 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 với 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

màn hình đơn sắc và màn hình màu. Ở Anh, quy định về an toàn và sức khoẻ năm 1992 yêu cầu các màn hình phải rõ ràng, dễ đọc và ổn định dưới điều kiện làm việc bình thường. Các yêu cầu trong phần này được mô tả như các đặc tả hiệu suất và chuẩn cung cấp các phương thức kiểm tra và các độ đo sự thích nghi. Các đặc tả bao gồm: Thiết kế khoảng cách quan sát; line-of-sight angle; góc quan sát; chiều cao ký tự ; độ rộng phím nhấn; tỉ lệ giữa chiều cao và chiều rộng của ký tự, định dạng ký tự,...

AUDIENCE: các nhà sản xuất màn hình và những người cần định giá chất lượng màn hình. Đây là một chuẩn kỹ thuật mà đòi hỏi phải có kiến thức về thiết kế màn hình và thị giác người.

#### 4. Keyboard requirements

CITATION: ISO 9241-4:1998 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

<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: cung cấp hướng dẫn về thiết kế các bàn phím cho các nghiệp vụ văn phòng. Các đặc tả bao gồm thiết kế chung của bàn phím và thiết kế của các phím. Thiết kế chung bao gồm các mục như: palm rest; bề mặt hiển thị của các phím; độ nghiêng của bàn phím và sự điều chỉnh độ nghiêng đó. Thiết kế các phím bao gồm: sơ đồ phím; sự dịch chuyển phím và lực tác dụng lên phím; phản hồi khi bấm phím; sự nảy lại khi nhấn phím; phím cuộn; chức năng lặp khi phím; chữ khắc trên phím; các phím mũi tên; phím phím số; và bề mặt trên của phím. Chuẩn này xác định các độ đô kỹ thuật cần để thẩm định việc tuân theo chuẩn, và bao gồm một bài kiểm tra hiệu suất người dùng cho các nhà sản xuất bàn phím mới.

AUDIENCE: các nhà sản xuất bàn phím và những người cần đánh giá chất lượng bàn phím.

#### 5. Workstation layout and postural requirements

CITATION: ISO 9241-5:1998 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 xác định các nguyên tắc hướng dẫn lao động học áp dụng cho thiết kế các thiết bị máy trạm sử dụng cho việc hiển thị các công việc màn hình. Chuẩn này nhấn mạnh rằng thiết kế không gian làm việc cần phải thực hiện sau khi phân tích các nhiệm vụ mà không gian làm việc đó hỗ trợ. Thông tin này sẽ giúp cho việc xác định các nhiệm vụ khác nhau được thực hiện và sự đóng góp tương đối của các thành phần khác nhau hỗ trợ cho các nhiệm vụ đó, nó sẽ giúp các người dùng chấp nhận tư thế làm việc có lợi cho sức khoẻ và thoải mái. Các đặc tả bắt nguồn từ 5 quy tắc hướng dẫn sau: versatility-flexibility; fit; postural change; user information; và maintainability-adapability. Bản thân các đặc tả bao gồm: tư thế; việc dễ dàng điều chỉnh; support surfaces; seating; additional support elements; và không gian làm việc. Một phụ lục thông tin cung cấp các thông tin về con người cần thiết cho việc thiết kế trạm làm việc và lựa chọn.

**AUDIENCE:** các nhà sản xuất các sản phẩm văn phòng và những người quan tâm đến việc thiết kế và cài đặt các văn phòng.

#### **6. Guidance on the work environment**

**CITATION:** ISO 9241-6:1999 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**

<input type="checkbox"/> General guidance	<input type="checkbox"/> Equipment	<input checked="" type="checkbox"/> Environment
<input type="checkbox"/> Software	<input type="checkbox"/> User Testing	<input type="checkbox"/> 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 đèn màn hình hiển thị. Các đặc tính của môi trường làm việc gồm sáu tinh 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**

<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**

Specifications  Principles and general

**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 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**

điều khiển sự phản xạ và ánh hường trên chất lượng ảnh; và mối quan hệ với ánh sáng môi trường. Chuẩn bao gồm một phần chi tiết mô tả cách thức đo và tính các độ phản xạ. Nó cũng bao gồm 4 phụ lục: một sự lựa chọn thay thế, phương thức kiểm tra dựa trên hiệu suất; định dạng báo cáo đồ họa; một thí dụ đã làm và một thư viện.

AUDIENCE: Đây là một chuẩn kỹ thuật dành cho những người có kiến thức sâu về light measurement. Đối tượng chủ yếu của tài liệu này là những kỹ sư màn hình, những người muốn chắc chắn rằng việc giảm sự phản xạ không làm giảm chất lượng hình ảnh.

### **8. Requirements for displayed colours**

CITATION: ISO 9241-8:1997 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

<input type="checkbox"/> Analyse the opportunity	<input checked="" 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 checked="" type="checkbox"/> Software	<input checked="" type="checkbox"/> User Testing	<input type="checkbox"/> Process

SCOPE: phần này xác định cách thức các màu sắc nên được sử dụng trên màn hình. Nó giới hạn các ứng dụng văn bản và đồ họa nhưng bao gồm cả 2 khía cạnh về giác quan và nhận biết màu sắc. Nguyên tắc hướng dẫn chính đó là: màu sắc nên được sử dụng để giúp cho con người quan sát, nhận biết và hiểu các đối tượng hình ảnh, thông tin một cách đúng đắn. Các đặc tả bao gồm: thiết lập màu mặc định; sự đồng đều về màu sắc; phân tán màu; chiều cao ký tự và kích thước đối tượng; sự khác nhau giữa các màu sắc; sự tương phản rõ ràng giữa ký tự và biểu tượng; spectrally extreme colours; nền và các hiệu ứng ánh bao quanh; và số lượng các màu. Chuẩn bao gồm các phần đặc tả về sự đo màu từ màn hình. Có 3 phụ lục: tính sự khác nhau giữa các màu; một kiểm tra tính hiệu quả của trực quan và một thư viện.

AUDIENCE: các nhà sản xuất màn hình (đặc biệt là màn hình CRT) và những người phát triển giao diện người dùng sử dụng màu để mã hoá, định dạng và thiết kế. Một số phần có tính kỹ thuật và yêu cầu phải có kiến thức về độ đo màu sắc.

## *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

<input type="checkbox"/> Analyse the opportunity	<input checked="" 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
--	---

#### 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 áp dụng cho vài kiều thiết bị ngoại vi như: chuột, pucks, joysticks, trackballs, tablets và overlays; touch-sensitive screens, stylus 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). Sứ 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

<input type="checkbox"/> Analyse the opportunity	<input checked="" type="checkbox"/> Create the user experience
<input type="checkbox"/> Build the context of use	<input type="checkbox"/> Track usage and improve

## **Phụ lục C: Các chuẩn trong thiết kế HCI**

### **TYPE OF GUIDANCE**

Specifications       Principles and general recommendations

### **APPLICATION AREA**

<input checked="" type="checkbox"/> General guidance	<input type="checkbox"/> Equipment	<input type="checkbox"/> Environment
<input checked="" type="checkbox"/> Software	<input type="checkbox"/> User Testing	<input type="checkbox"/> 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 tuần 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**

<input checked="" type="checkbox"/> Analyse the opportunity	<input checked="" type="checkbox"/> Create the user experience
<input checked="" type="checkbox"/> Build the context of use	<input checked="" type="checkbox"/> Track usage and improve

### **TYPE OF GUIDANCE**

Specifications       Principles and general recommendations

### **APPLICATION AREA**

<input type="checkbox"/> General guidance	<input checked="" type="checkbox"/> Equipment	<input type="checkbox"/> Environment
<input checked="" type="checkbox"/> Software	<input checked="" type="checkbox"/> User Testing	<input checked="" type="checkbox"/> 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

cụ thể để thực hiện nhiệm vụ hiệu quả, hiệu suất và thỏa mãn 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**

<input type="checkbox"/>	Analyse the opportunity	<input checked="" type="checkbox"/>	Create the user experience
<input 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 type="checkbox"/>	General guidance	<input type="checkbox"/>	Equipment	<input type="checkbox"/>	Environment
<input checked="" type="checkbox"/>	Software	<input checked="" type="checkbox"/>	User Testing	<input type="checkbox"/>	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

<input type="checkbox"/> Analyse the opportunity	<input checked="" type="checkbox"/> Create the user experience
<input 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 type="checkbox"/> General guidance	<input type="checkbox"/> Equipment	<input type="checkbox"/> Environment
<input checked="" type="checkbox"/> Software	<input type="checkbox"/> User Testing	<input type="checkbox"/> 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

<input type="checkbox"/> Analyse the opportunity	<input checked="" type="checkbox"/> Create the user experience
<input 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 type="checkbox"/> General guidance	<input type="checkbox"/> Equipment	<input type="checkbox"/> Environment
<input checked="" type="checkbox"/> Software	<input type="checkbox"/> User Testing	<input type="checkbox"/> Process

## **Phụ lục C: Các chuẩn trong thiết kế HCI**

---

**SCOPE:** phần này cung cấp các khuyến nghị về thiết kế của các hệ thống có sử dụng menus. Chuẩn bắt đầu bằng việc duyệt lại các lĩnh vực ứng dụng có sử dụng menu nhiều. Các khuyến nghị bao gồm: cấu trúc menu (như là logical categories, grouping options và ordering items); menu navigation (bao gồm các tiêu đề và thời gian truy cập); lựa chọn và thực thi các tùy chọn (bao gồm các phương thức lựa chọn, sử dụng bàn phím và âm thanh); menu presentation (bao gồm sự thay thế và sử dụng các icons). Chuẩn bao gồm ba phụ lục. Đầu tiên là một thủ tục mẫu để thẩm định khả năng áp dụng được và tôn trọng, và bao gồm 10-page checklist. Thứ hai là phụ lục cung cấp ba kịch bản về cách áp dụng chuẩn (tù quan điểm của người thiết kế giao diện người dùng, người mua và người đánh giá). Phụ lục thứ ba là thư viện chi tiết.

**AUDIENCE:** user interface designers and evaluators.

### **15. Command Dialogues**

**CITATION:** ISO 9241-15:1997 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**

<input type="checkbox"/> Analyse the opportunity	<input checked="" type="checkbox"/> Create the user experience
<input 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 type="checkbox"/> General guidance	<input type="checkbox"/> Equipment	<input type="checkbox"/> Environment
<input checked="" type="checkbox"/> Software	<input type="checkbox"/> User Testing	<input type="checkbox"/> Process

**SCOPE:** phần này cung cấp các khuyến nghị cho các hệ thống có sử dụng giao diện dòng lệnh (như DOS hay UNIX). Với giao diện dòng lệnh, người dùng làm việc với hệ thống bằng cách gõ vào các lệnh theo một quy tắc cú pháp nào đó. Chuẩn bắt đầu bằng việc mô tả các miền ứng dụng thích hợp cho các giao diện này (ví dụ, các ứng dụng mà con người sử dụng thường xuyên, và yêu cầu tốc độ cũng như sự mềm dẻo). Các khuyến nghị bao gồm: ngữ pháp và cấu trúc (macro, command arguments,...); biểu diễn lệnh (tên lệnh và viết tắt); input and output considerations (xem xét vào ra,...); phản hồi và trợ giúp (xử lý lệnh và phản hồi lỗi,...); Chuẩn có hai phụ lục. Thứ nhất là một thủ tục mẫu để đánh giá khả năng áp dụng được, tôn trọng và bao gồm 5-page checklist. Phụ lục thứ hai là một thư viện.

**AUDIENCE:** user interface designers và evaluators.

### **16. Direct manipulation dialogues**

CITATION : ISO 9241-16:1999 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**

<input type="checkbox"/> Analyse the opportunity	<input checked="" type="checkbox"/> Create the user experience
<input 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 type="checkbox"/> General guidance	<input type="checkbox"/> Equipment	<input type="checkbox"/> Environment
<input checked="" type="checkbox"/> Software	<input type="checkbox"/> User Testing	<input type="checkbox"/> Process

**SCOPE:** phần này cung cấp các khuyến nghị cho các hệ thống mà sử dụng các thao tác trực tiếp (như Windows và Macintosh). Với thao tác trực tiếp, người sử dụng có thể tương tác trực tiếp với các đối tượng trên màn hình. Chuẩn này chỉ ra rằng nó không giống như giao diện đồ họa (có thể cài đặt hay không các thao tác trực tiếp). Chuẩn này trước tiên mô tả miền ứng dụng tương ứng cho các giao diện này (hệ thống có thể mô phỏng các đối tượng tác vụ trong thế giới thực, các thuộc tính của nó và các thao tác). Khuyến nghị bao gồm: thông tin chung (cảnh trí-metaphors, giao diện của các đối tượng sử dụng trong thao tác; phản hồi và các thiết bị nhập); thao tác của các đối tượng, thao tác trực tiếp của các đối tượng văn bản, thao tác trực tiếp của các cửa sổ và thao tác trực tiếp của điều khiển icons. Chuẩn chứa một phụ lục và thư viện bibliography. Phụ lục là một thủ tục mẫu để đánh giá khả năng áp dụng được, tôn trọng và bao gồm 3-page checklist.

**AUDIENCE:** user interface designers and evaluators.

### **17. Form filling dialogues**

CITATION: ISO 9241-17:1998 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**

<input type="checkbox"/> Analyse the opportunity	<input checked="" type="checkbox"/> Create the user experience
<input type="checkbox"/> Build the context of use	<input type="checkbox"/> Track usage and improve

## **Phụ lục C: Các chuẩn trong thiết kế HCI**

### **TYPE OF GUIDANCE**

Specifications       Principles and general recommendations

### **APPLICATION AREA**

<input type="checkbox"/> General guidance	<input type="checkbox"/> Equipment	<input type="checkbox"/> Environment
<input checked="" type="checkbox"/> Software	<input type="checkbox"/> User Testing	<input type="checkbox"/> Process

**SCOPE:** phần này cung cấp các khuyến nghị cho các hệ thống sử dụng các giao diện dạng form điện. Với ứng dụng của chúng trên màn hình, 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 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

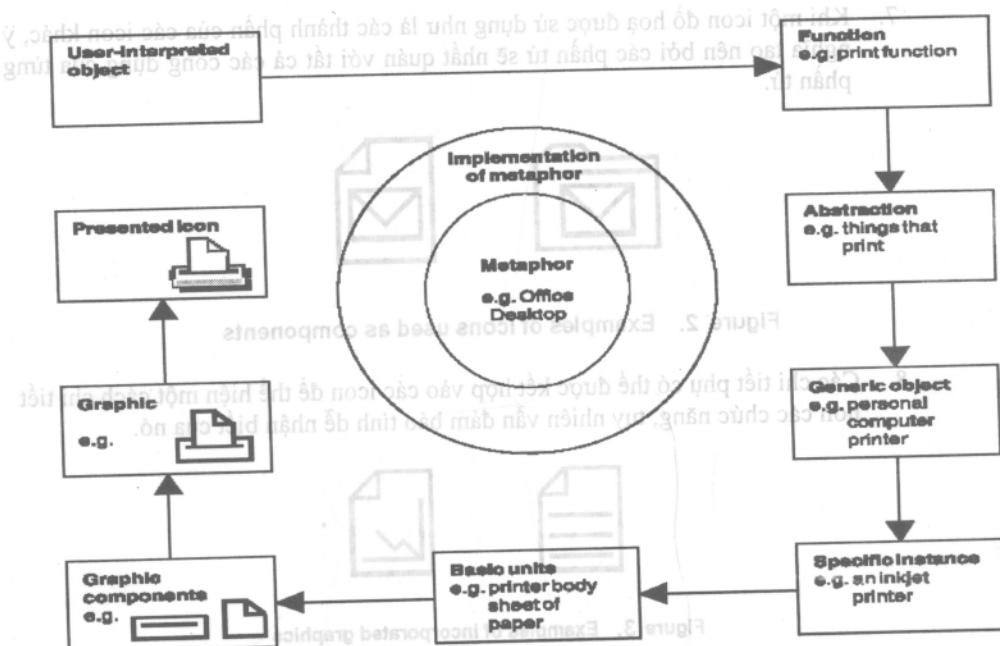


Figure 1. Framework used to specify icons.

### 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 đè lên 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 xoá 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.

## *Phụ lục C: Các chuẩn trong thiết kế HCI*

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.

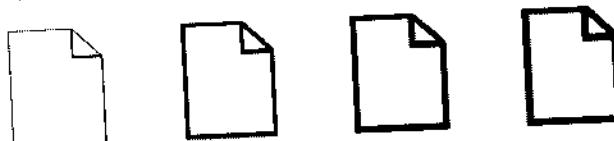


Figure 4. Examples of variations in the line attribute "width".

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.

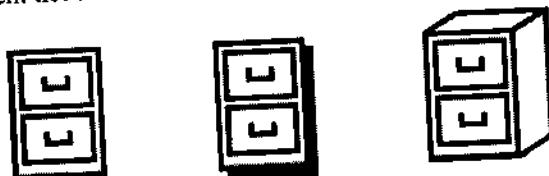


Figure 6. Examples of levels of detail

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. Mở 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.



Basic object icon



Examples of different states

**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

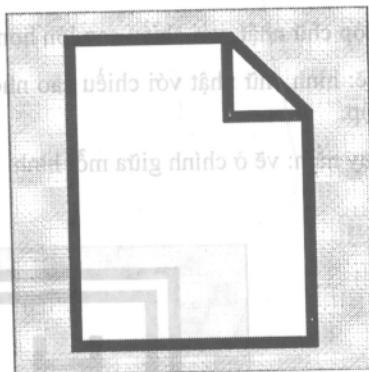
- Biểu tượng đối tượng khung: có thể chấp nhận các đối tượng khác như là đầu vào.
- Biểu tượng đối tượng thứ cấp: không thể chấp nhận các đối tượng khác là đầu vào. Có hai dạng:
  - + Biểu tượng đối tượng phụ.
  - + Biểu tượng đối tượng môi trường hệ thống.

- Biểu tượng đối tượng khung:

1. Văn bản

- o 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.
- o 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.

- o Hình



2. Thư mục (Folder):

- o 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.)
- o Thể hiện cụ thể: Một thùng chứa - a manila folder
- o Các thành phần:
  - Mặt sau: hình chữ nhật mà chiều cao nhỏ hơn chiều rộng và có một đường kẻ dọc theo đỉnh.

- Mặt trước: hình chữ nhật với chiều cao nhỏ hơn chiều rộng và định vị ở phía trước đối với mặt sau.

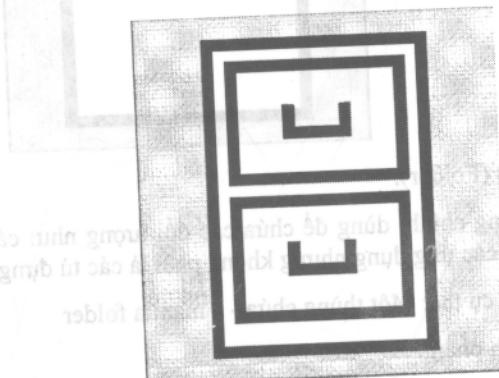
- Hình**



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.

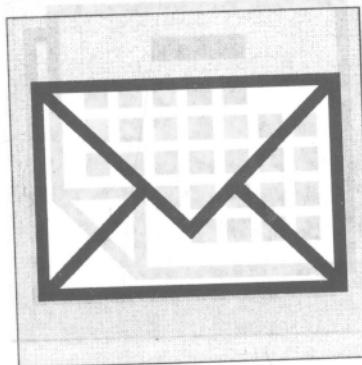
- Hình**



## **Phụ lục C: Các chuẩn trong thiết kế HCI**

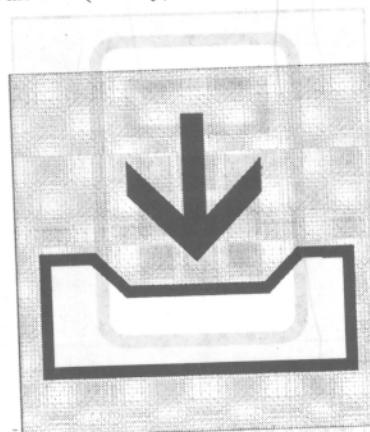
### 4. Phong bì

- Hình



### 5. Tủ đựng công văn đến (In-tray)

- Hình



Ngoài ra còn có nhiều các đối tượng khác như tủ đựng công đèn, máy in, điện thoại,...

#### • Các biểu tượng đối tượng khác

##### 1. Lịch

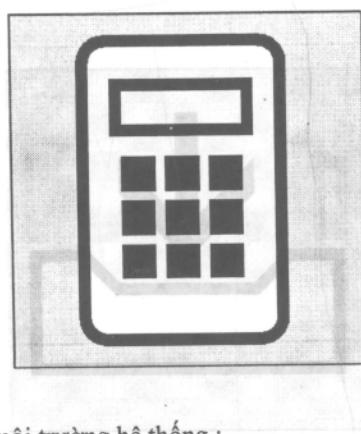
- Chức năng chính: cho phép thực hiện chức năng ghi và lập lịch các sự kiện
- Thể hiện cụ thể: Một lịch giấy
- Các thành phần:
  - Khung lịch:
  - Chi tiết các tháng:

- o Hình



2. Máy tính bỏ túi

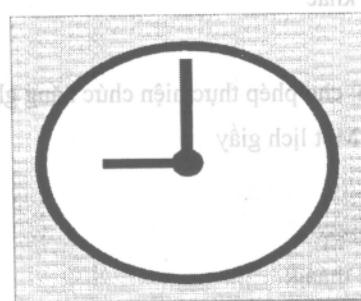
- o Hình



- Biểu tượng đối tượng môi trường hệ thống :

1. Đồng hồ

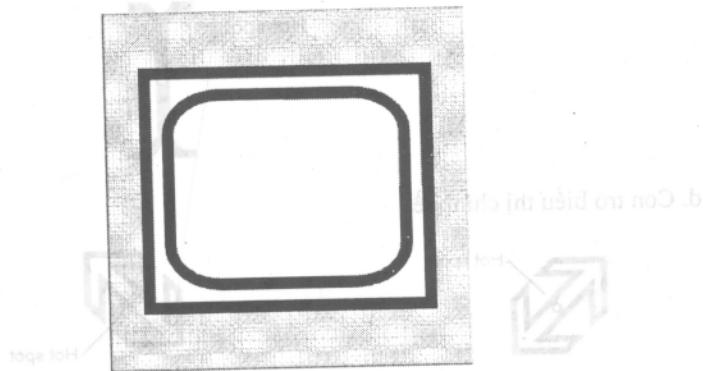
- o Hình



## **Phụ lục C: Các chuẩn trong thiết kế HCI**

### 2. Thiết bị hiển thị

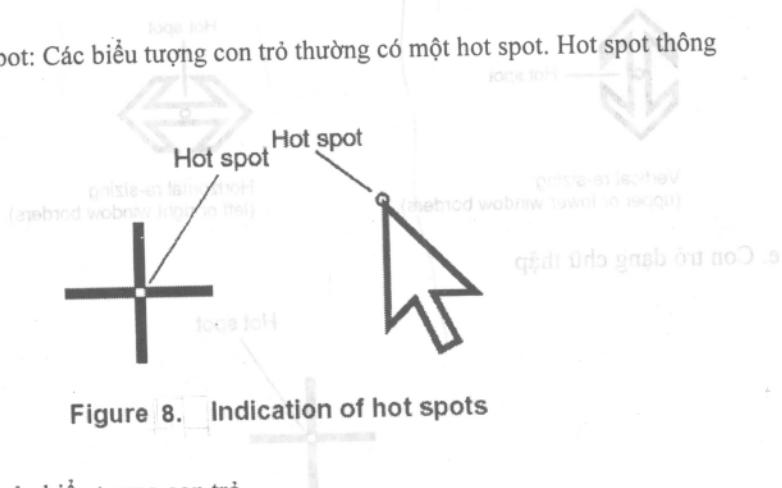
#### o Hình



### 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.

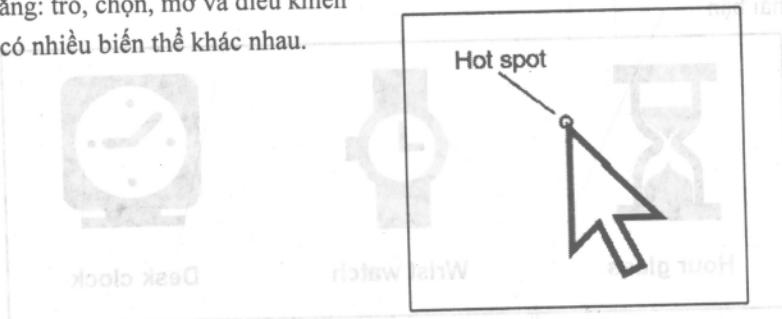


**Figure 8. Indication of hot spots**

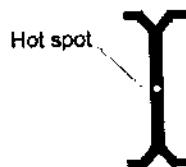
b. Dạng ngầm định của biểu tượng con trỏ

Chức năng: trỏ, chọn, mở và điều khiển

Chúng có nhiều biến thể khác nhau.



c. Con trỏ định vị văn bản:



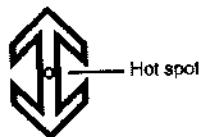
d. Con trỏ biểu thị chỉnh lề



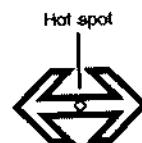
Horizontal and vertical re-sizing  
(upper left and lower right window corners).



Horizontal and vertical re-sizing  
(upper right and lower left window corners).

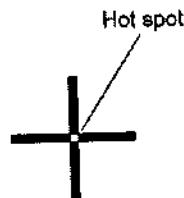


Vertical re-sizing  
(upper or lower window borders).



Horizontal re-sizing  
(left or right window borders).

e. Con trỏ dạng chữ thập



f. Chỉ trạng thái bận

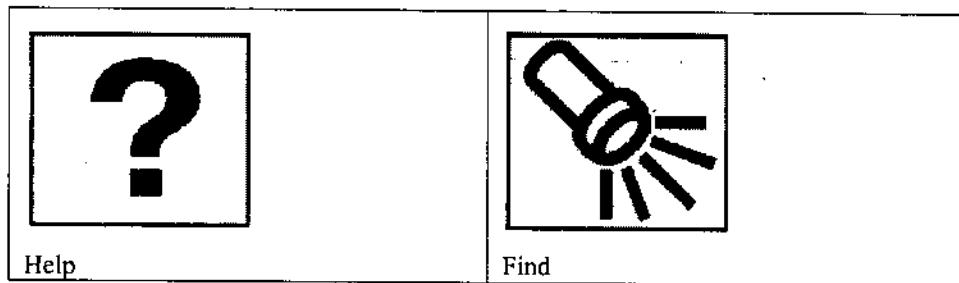


#### 4. Biểu tượng hành động -Action Icons

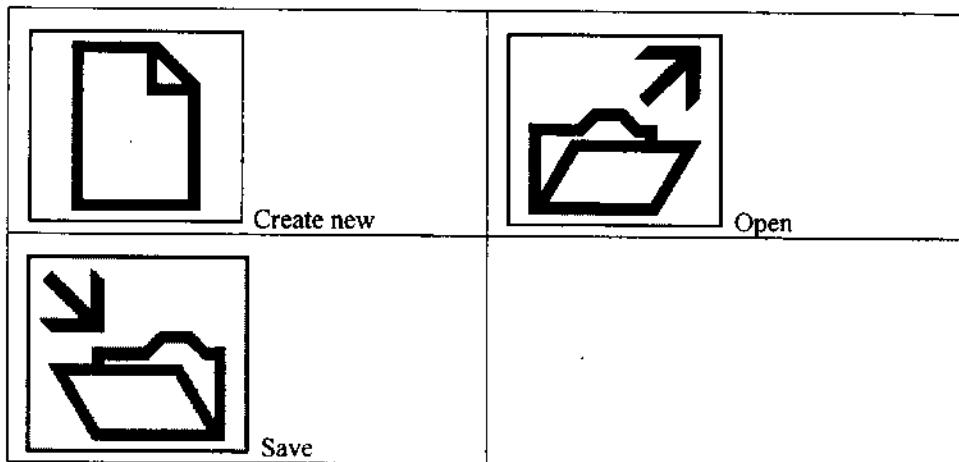
Biểu tượng hành động được phân thành sáu nhóm:

- Hành động chung.
- Hành động điền.
- 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

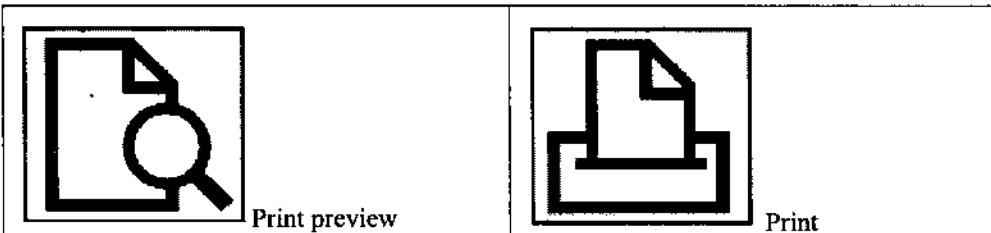


##### b. Hành động điền

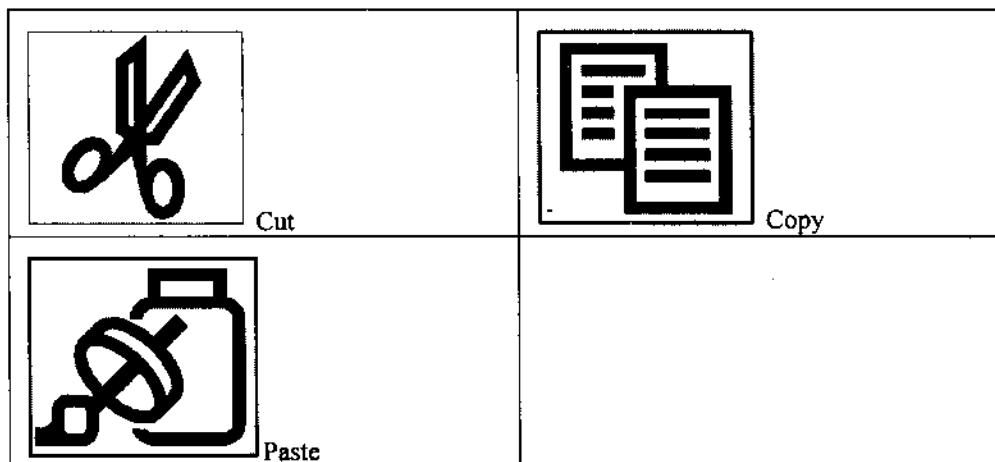


### **Phụ lục C: Các chuẩn trong thiết kế HCI**

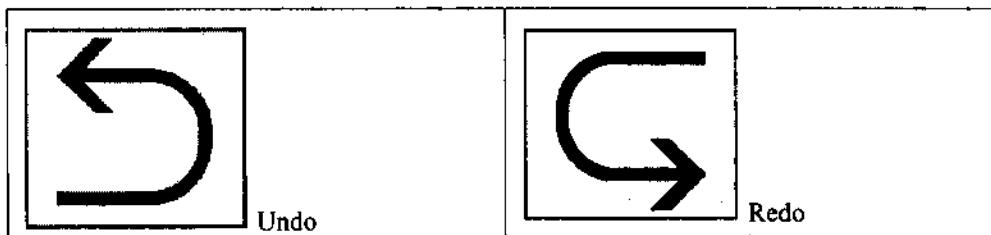
#### c. Hành động in



#### d. Hành động cơ bản của soạn thảo

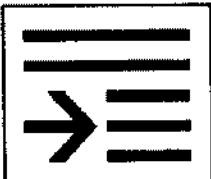
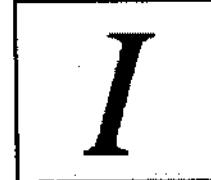
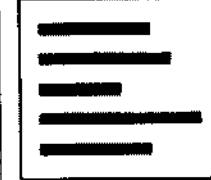
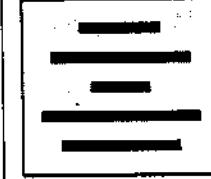
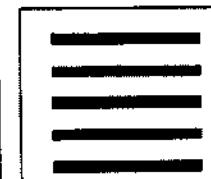


#### e. Hủy bỏ hay khôi phục hành động



### *Phụ lục C: Các chuẩn trong thiết kế HCI*

#### f. Hành động với văn bản

	Increase indent		Decrease indent
	Enumerate/numbering		Itemise
	Make Bold		Italicise
	Underline		Align left
	Align right		Centralize
	Full justify		

## 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-ordinareur, 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 conviale,* 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 (User 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] *Nigel Bevan, Human-Computer Interaction Standards, Proceedings of the 6<sup>th</sup>  
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.

## Tài liệu tham khảo

---

- [18] *Lương Mạnh Bá, Huỳnh Quyết Thắng, Lê Tân Hùng*, Tương tác người - máy. Ứng dụng kỹ thuật đồ họa và công nghệ 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 kỵ yếu của Hội nghị.
- [20] Webdesign.about.com/library.
- [21] Usable web.com/authors/
- [22] <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**

**NGỌC LINH**

*Trình bày bìa:*

**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**