

**Lời cảm ơn :**

Đầu tiên, chúng em xin cảm ơn Bộ môn Công Nghệ Phần Mềm - Khoa Công nghệ thông tin đã hỗ trợ về tài liệu, máy móc cho chúng em thực hiện khóa luận này. Chúng em cũng thực sự cảm ơn Thầy Lê Hoài Bắc đã hướng dẫn tận tình trong suốt quá trình thực hiện khoá luận. Và cuối cùng, chúng em cảm ơn các thầy cô và các bạn trong Khoa đã ủng hộ và nhiệt tình giúp đỡ cho chúng em hoàn thành khoá luận.

*Ngày 20 tháng 07 năm 2005*

## **Mục lục :**

Phần giới thiệu .....	7
Tóm tắt luận văn .....	9
<b>Chương 1 Giới thiệu về ITS (Intelligent Tutoring Systems).....</b>	<b>10</b>
1.1/ Định nghĩa .....	10
1.2/ Mục đích.....	11
1.3/ Sơ lược về lịch sử hình thành .....	12
1.3.1/ Trí tuệ nhân tạo (AI- Artificial Intelligence) .....	12
1.3.2/ Những hệ thống hướng dẫn hỗ trợ dựa máy tính.....	12
1.3.3/ Sự rèn luyện và đào sâu thích hợp .....	13
1.3.4/ Sự khủng hoảng của Trí tuệ nhân tạo và Tâm lý học giáo dục .....	14
1.3.5/ Trí tuệ nhân tạo và Hệ thống dạy học thông minh (AI – ITS).....	14
1.3.6/ Vai trò của khoa học nhận thức .....	16
1.4/ Những thách thức đối với các hệ thống ITS.....	18
1.5/ Sơ lược về kiến trúc của một hệ thống ITS .....	22
1.5.1/ Mô hình lĩnh vực (Tri thức về lĩnh vực ) .....	23
1.5.2/ Mô hình người học.....	27
1.5.3/ Mô hình dạy học .....	29
1.5.4/ Giao diện của hệ thống .....	30
1.6/ Kết luận .....	32
<b>Chương 2 Những công cụ soạn thảo ITS.....</b>	<b>33</b>
2.1/ Giới thiệu .....	33
2.2/ Công cụ soạn thảo ITS - Mục đích .....	34
2.2.1/ Công cụ soạn thảo ITS .....	34
2.2.2/ Mục đích .....	38
2.3/ So sánh giữa ESS (Expert System Shells) và ITSAT .....	39
2.4/ Những yêu cầu của một ITSAT .....	42
2.4.1/ Tái sử dụng .....	42
2.4.2/ Kiểm soát hệ thống .....	45

2.4.3/	Những chú thích bằng hình ảnh cho sự đạt được tri thức .....	47
2.4.4/	Phát triển những giao diện lĩnh vực .....	48
2.4.5/	Tính tổng quát và riêng biệt .....	50
2.5/	Dạy học những kỹ năng thủ tục .....	53
2.6/	Giới thiệu một vài công cụ soạn thảo .....	55
2.6.1/	Sự phân loại theo tác vụ và dạy học .....	56
2.6.1.1/	Sự sắp thứ tự và lên kế hoạch môn học .....	60
2.6.1.2/	Những chiến lược dạy học .....	61
2.6.1.3/	Giả lập thiết bị và huấn luyện thiết bị.....	62
2.6.1.4/	Hệ chuyên gia lĩnh vực .....	63
2.6.1.5/	Loại đa tri thức .....	64
2.6.1.6/	Mục đích đặc biệt .....	65
2.6.1.7/	Chương trình dạy học dựa trên media thích ứng / thông minh .....	66
2.6.2/	Soạn thảo nội dung của một ITS .....	66
2.6.2.1/	Soạn thảo giao diện .....	67
2.6.2.2/	Soạn thảo mô hình lĩnh vực .....	68
2.6.2.2.1/	Những mô hình của tri thức và cấu trúc lĩnh vực.....	68
2.6.2.2.2/	Sự giả lập và những mô hình .....	69
2.6.2.2.3/	Những mô hình của chuyên môn lĩnh vực .....	69
2.6.2.2.4/	Loại tri thức lĩnh vực .....	70
2.6.2.3/	Soạn thảo mô hình dạy học .....	70
2.6.2.3.1/	Những hệ thống dựa kế hoạch .....	71
2.6.2.3.2/	Đa chiến lược.....	71
2.6.2.3.3/	Từ vựng cho hoạt động dạy học .....	72
2.6.2.4/	Soạn thảo mô hình người học.....	72
2.6.3/	Những phương pháp soạn thảo và phương pháp đạt được tri thức .....	72
2.7/	Kết luận .....	73
<b>Chương 3</b>	<b>Phát triển ITS sử dụng Cấu trúc tri thức.....</b>	<b>75</b>
3.1/	Giới thiệu.....	75

3.2/ Phạm vi lĩnh vực .....	76
3.3/ Phương pháp luận khái niệm hoá .....	77
3.4/ Cấu trúc Tri thức .....	80
3.4.1/ Định nghĩa .....	80
3.4.2/ Tính chất .....	80
3.4.3/ Các Cấu trúc tri thức .....	81
3.4.3.1/ Cấu trúc Tri thức 1 .....	81
3.4.3.2/ Cấu trúc Tri thức 2 .....	82
3.4.3.3/ Cấu trúc Tri thức 3 .....	84
3.4.3.4/ Cấu trúc Tri thức 4 .....	85
3.4.3.5/ Cấu trúc Tri thức 5 .....	86
3.4.3.6/ Cấu trúc Tri thức 6 .....	87
3.5/ Các đối tượng chính được thực hiện trên Cấu trúc tri thức.....	89
3.5.1/ Danh mục các đối tượng đơn giản .....	89
3.5.2/ Sự tuần tự của các đối tượng .....	89
3.5.3/ Cặp các đối tượng .....	89
3.6/ Những người sử dụng của hệ thống .....	89
3.7/ Kiến trúc của hệ thống .....	92
3.7.1/ Cấu trúc tri thức ( Tri thức lĩnh vực ) .....	93
3.7.2/ Bộ giả lập .....	94
3.7.3/ Bộ soạn thảo .....	97
3.7.4/ Bộ đánh giá .....	99
3.8/ Sự mô tả xử lý bên trong hệ thống .....	100
<b>Kết luận và Hướng phát triển .....</b>	<b>102</b>
<b>Phụ lục .....</b>	<b>107</b>

**Bảng các hình vẽ :**

<b>Ký hiệu</b>	<b>Tên hình vẽ</b>
H1.1	Các thành phần của hệ thống ITS
H1.2	Những vấn đề trong việc phát triển hệ thống ITS
H1.3	Mô hình ITS thông thường
H1.4	Biểu diễn tri thức trên InfoMap
H1.5	Một định nghĩa XML Schema cho một thực thể trong Mô hình lĩnh vực
H1.6	Màn hình chính của hệ thống CIRCSIM-Tutor
H2.1	Tổng quan về môi trường phát triển ITS
H2.2	Sự phân chia kỹ thuật biểu diễn và tri thức lĩnh vực
H2.3	Những thành phần với sự phụ thuộc và sự độc lập lĩnh vực
H2.4	Chia sẻ tri thức lĩnh vực độc lập
H2.5	Hai hướng tiếp cận đến phát triển giao diện
H2.6	Sự phân chia giao diện ITS
H2.7	Phân chia những thành phần ITS
H3.1	Theo một tiêu chuẩn được chọn trước
H3.2	Tổ chức các thành phần tạo thành các danh mục
H3.3	Hình thành sự tuần tự
H3.4	Hoàn chỉnh sự tuần tự
H3.5	Thiết lập các mối liên kết chính xác
H3.6	Thiết lập đa liên kết
H3.7	Giáo viên trong vai trò là “người xây dựng bài tập”
H3.8	Giáo viên trong vai trò là “người sử dụng”
H3.9	Kiến trúc hệ thống và Sự tương tác giữa người sử dụng và hệ thống
H3.10	Cấu trúc tri thức gắn với giao diện soạn thảo
H3.11	Giao diện giả lập bài tập sử dụng Cấu trúc tri thức về sự tuần tự

H3.12	Giao diện giả lập bài tập sử dụng Cấu trúc tri thức về sự liên kết
H3.13	Trình soạn thảo cho đối tượng với Cấu trúc tri thức tuần tự
H3.14	Trình soạn thảo cho việc xác định các đối tượng âm thanh với Cấu trúc tri thức về sự liên kết
H3.15	Quá trình xử lý và người dùng trong hệ thống
H3.16	Màn hình giả lập bài tập Dạng đầu tiên mà Trẻ làm
H3.17	Từ lĩnh vực đến bài tập

**Bảng các Bảng :**

Ký hiệu	Tên bảng
Bảng 1.1	Những giả định ACT* và những nguyên tắc liên quan cho việc dạy học được thực thi trên máy tính.
Bảng 2.1	Danh mục phân loại những ITSAT
Bảng 2.2	Sức mạnh và giới hạn của những ITSAT theo danh mục

**Bảng các ký hiệu và chữ viết tắt :**

Ký hiệu	Tên ký hiệu hoặc chữ viết tắt
ITS	Intelligent Tutoring System
ITSAT	Intelligent Tutoring System Authoring Tool
CAI	Computer Aided Instruction
CBT	Computer-Based Training
AI	Artificial Intelligence
ESS	Expert System Shells
Ius	Instruction Units

**Phần giới thiệu :**

Giáo dục là một trong những lĩnh vực được áp dụng sớm nhất của máy tính. Nhiều hệ thống dạy học đã ra đời để đáp ứng nhu cầu học tập của con người. Tuy thật phát triển nhưng người ta thấy rằng những hệ thống này chỉ đơn thuần là “dạy” mà không có bất kỳ chiến lược dạy học và hướng dẫn nào cho từng loại người học khác nhau, cũng như không đánh giá được quá trình học của người học. Từ đó những hệ thống dạy học thông minh (ITS – Intelligent Tutoring System) đã ra đời dựa trên những hệ thống dạy học truyền thống mà được thêm vào các thành phần “thông minh” trên.

Những hệ thống ITS là những hệ thống đang được nghiên cứu nhiều trên thế giới. Người ta hi vọng rằng chúng có thể thay thế những hệ thống dạy học truyền thống. Tuy đây là một hướng tiếp cận đầy tiềm năng nhưng do gặp nhiều khó khăn không lường trước nên những hệ thống ITS đã phát triển không như mong đợi. Vì vậy chúng chỉ mới đang ở giai đoạn đầu phát triển.

#### **Mục tiêu :**

Nghiên cứu những hệ thống này, khóa luận chỉ muốn đạt được những mục tiêu sau :

- Trình bày thế nào là hệ thống dạy học thông minh (ITS)?
- Kiến trúc và cách thức hoạt động của một hệ thống ITS?
- Những hướng tiếp cận để xây dựng hệ thống ITS?
- Cuối cùng sẽ áp dụng trong một lĩnh vực cụ thể.

#### **Nội dung trình bày :**

Như những mục tiêu đã đề ra ở trên, khóa luận sẽ nghiên cứu những vấn đề sau :

- Trình bày khái niệm, kiến trúc và cách thức hoạt động cơ bản của hệ thống ITS.
- Trình bày một hướng tiếp cận mới để xây dựng hệ thống ITS: đó là công cụ soạn thảo ITS (ITSAT – ITS Authoring Tool).

- Trong phần áp dụng : khác với phần mục tiêu đã đề ra, mục đích chính trong phần này mà khóa luận muốn trình bày là :
  - Nghiên cứu và xây dựng một phương pháp để biểu diễn tri thức lĩnh vực.
  - Xây dựng ITS thành một công cụ hỗ trợ giáo viên xây dựng bài tập dựa trên những tri thức trên.
- Cùng với 2 mục đích chính trên, khóa luận cũng sẽ trình bày thêm những phần tạo thành một hệ thống ITS hoàn chỉnh.

### **Kết quả :**

- Về phần lý thuyết, khóa luận trình bày nghiên cứu đầy đủ về hệ thống ITS như trong phần nội dung đã nêu.
- Về phần áp dụng : mục đích của khóa luận không phải là nghiên cứu và xây dựng một hệ thống ITS hoàn chỉnh mà mục đích chính của như trình bày trong phần nội dung là xây dựng ITS thành một công cụ giúp cho giáo viên – những người không cần biết lập trình – có thể soạn thảo bài tập bằng cách sử dụng những tri thức đã được lưu trữ trong hệ thống cùng với một số thành liên quan.

### **Tóm tắt khóa luận :**



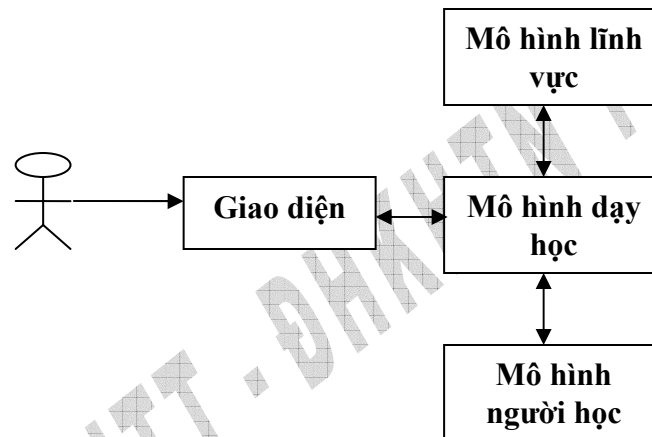
Khóa luận này nghiên cứu về những hệ thống hỗ trợ dạy học thông minh (Intelligent Tutoring Systems – ITS). Đây là những hệ thống dạy học hiện đại (so với các phần mềm dạy học truyền thống) đã và đang được nghiên cứu nhiều trong các trường đại học và phòng thí nghiệm trên thế giới. Và mục đích chính của khóa luận này là sẽ nghiên cứu và xây dựng một hệ thống ITS cho một lĩnh vực được chọn. Đây là một công cụ được xây dựng để hỗ trợ cho người dạy (giáo viên...) mà không cần phải biết lập trình có thể phác thảo ra các bài tập giảng dạy về một lĩnh vực nào đó dựa trên máy tính. Người dạy tạo ra những bài tập trên bằng cách sử dụng Tri thức được lưu trữ trong máy tính, chọn ra những nội dung phù hợp theo yêu cầu để hệ thống tự phát sinh. Ngoài ra hệ thống còn xây dựng một “bộ giả lập” để người học (trẻ ...) có thể tự rèn luyện những bài tập đã soạn bởi người dạy. Quá trình này được kiểm soát và đánh giá bởi người dạy và hệ thống ( cụ thể là Module hướng dẫn và Module đánh giá). Khóa luận được nghiên cứu thành 3 phần chính. Phần đầu sẽ trình bày tổng quan về các hệ thống dạy học thông minh (ITS) (Chương 1). Phần thứ hai khóa luận muốn mô tả những công cụ soạn thảo ITS (ITSAT) (Chương 2). Phần cuối của khóa luận sẽ nghiên cứu và xây dựng hệ thống ITS sử dụng Cấu trúc tri thức (Chương 3).

## **Chương 1 : Giới thiệu về ITS (Intelligent Tutoring Systems)**

### **1.1/ Định nghĩa :**

Song song với việc thiết kế hướng dẫn và phần mềm dạy học, hệ thống dạy học thông minh (ITS) đã được phát triển. ITS là hệ thống hướng dẫn dựa trên máy tính với những mô hình nội dung hướng dẫn xác định cái gì được dạy và những chiến lược dạy học xác định cách để dạy. ITS là một trong những đóng góp quan trọng của trí tuệ nhân tạo đối với giáo dục. Hầu hết ITS có lõi là một hệ chuyên gia. Mỗi ITS phải có 4 thành phần chính (Hình 1.1) :

- Mô hình của tri thức lĩnh vực (Mô hình lĩnh vực).
- Mô hình người học.
- Những chiến lược dạy học được mô hình hoá (Mô hình dạy học).
- Một thành phần giao tiếp giữa hệ thống và người học (Giao diện hệ thống) .



**H1.1: Các thành phần của hệ thống ITS**

## 1.2/ Mục đích :

Việc sử dụng máy tính như một công cụ giảng dạy sớm được xem xét trong sự phát triển của máy tính. Sự phát triển của “giáo viên nhân tạo” là lý tưởng của lĩnh vực này. Sự tự động của toàn bộ hệ thống giảng dạy sẽ cung cấp nhiều thuận lợi bao gồm lợi ích của việc dạy học 1-1, có hiệu quả về mặt chi phí, việc tái sử dụng và tiêu chuẩn hoá những nguồn tài nguyên và môi trường học mang tính cá nhân hơn. Thật không may, sự phức tạp của quá trình học và dạy đã làm cho ý tưởng này không thực hiện được.

Việc sử dụng hệ thống hỗ trợ dạy học dùng máy tính cung cấp nhiều thuận lợi hơn so với việc tiếp cận phương pháp giảng dạy truyền thống. Người học sẽ nhận được thông tin phản hồi ngay lập tức khi mà việc chấm điểm được thực hiện một cách tự động. Thêm vào đó, với việc huấn luyện dựa trên máy tính thì thời gian hướng dẫn sẽ phụ thuộc vào khả năng của cá nhân người đó chứ không như trong lớp học truyền thống thời gian này phụ thuộc vào khả năng của người học kém nhất lớp.

Đối với nhiều người học, môi trường học trên máy tính ít có tính nghiêm túc hơn là trong lớp học truyền thống.

Môi trường máy tính cung cấp những sự hướng dẫn trực tiếp và trong nhiều trường hợp việc học qua máy tính sẽ tốn ít chi phí hơn những thiết bị đặc biệt mắc tiền và nguy hiểm. Việc đào tạo chất lượng cao trở thành có giá trị ở những vùng xa và sự hướng dẫn có thể gửi đến người học vì thế người hướng dẫn sẽ không cần phải đi xa và trang thiết bị không cần phải vận chuyển.

Những nghiên cứu về hệ thống này đã chỉ ra rằng: người học được giảng dạy bởi ITS sẽ học nhanh hơn và tốt hơn những người học tham gia các lớp học chung.

**Ví dụ:** Ở trường đại học Carnegie Mellon, các nhà nghiên cứu đã phát triển một hệ thống ITS gọi là LISP Tutor vào giữa những năm 1980 để dạy những kỹ năng lập trình máy tính cho sinh viên. Một cuộc khảo sát đã cho thấy, 43% sinh viên sử dụng hệ thống ITS có số điểm cao hơn những sinh viên học theo lối truyền thống. Khi đưa ra

những vấn đề lập trình phức tạp những sinh viên học theo lối truyền thống sẽ tốn nhiều hơn 30% thời gian so với những sinh viên được học từ ITS.

Nhưng cũng tồn tại nhiều bất lợi trong môi trường huấn luyện được hỗ trợ của máy tính. Môi trường này không uyển chuyển bằng môi trường dạy học thông thường. Chế độ trao đổi thông tin giữa giáo viên và người học bị giới hạn. Những phản ứng của sinh viên phải nằm trong sự đoán trước của hệ thống. Những câu hỏi không mong đợi của người học chỉ nhận về những phản hồi không tương xứng.

### **1.3/ Sơ lược về lịch sử hình thành :**

#### **1.3.1/ Trí tuệ nhân tạo (AI- Artificial Intelligence) :**

Những hệ thống dạy học thông minh (ITS) có một lịch sử rất thú vị, bắt nguồn từ sự phát triển của Trí tuệ nhân tạo (AI) vào cuối thập niên 1950 và đầu thập niên 1960. Sau đó, các nhà nghiên cứu Trí tuệ nhân tạo như Alan Turing, Marvin Minsky, John McCarthy, và Allen Newell nghĩ rằng máy tính có thể “suy nghĩ như con người”. Rất nhiều người nghĩ rằng sự ràng buộc chính cho mục đích này là sự tạo ra những máy tính to hơn và nhanh hơn. Người ta dường như có lý để giả định rằng một khi chúng ta đã tạo ra những máy móc có thể suy nghĩ chúng có thể thực hiện bất cứ tác vụ nào mà chúng ta liên kết với suy nghĩ của con người, chẳng hạn sự hướng dẫn.

#### **1.3.2/ Những hệ thống hướng dẫn hỗ trợ dựa máy tính :**

Trong thập niên 1960, những nhà nghiên cứu đã tạo ra một khối lượng nhiều các hệ thống hướng dẫn hỗ trợ dựa máy tính mà có khả năng phát sinh. Những chương trình này phát sinh các tập các vấn đề được thiết kế để nâng cao hiệu suất người học trong những lĩnh vực dựa kỹ năng, đại số cơ bản, và khả năng nhớ lại từ vựng. Về cơ bản, đây là những hệ thống tự phát sinh, được thiết kế để trình bày cho người học một vấn đề, nhận và ghi lại những phản hồi của người học, và lập bảng kê tất cả khả năng của người học về tác vụ đó. Hầu hết những nỗ lực của người thiết kế hệ thống được

dành cho sự mót hàm với những thách thức kĩ thuật của việc lập trình những hệ thống này trên các máy tính tương đối công kênh và đắt đỏ về thời gian. Những hệ thống này không cho ra những phiên bản một cách tường minh của cách mà người ta học, với một mô hình chuyển đổi / hành vi cư xử không rõ ràng của việc dạy và việc học. Họ giả định rằng nếu những hệ thống trình bày thông tin cho người học, người học sẽ hấp thu nó.

### **1.3.3/ Sự rèn luyện và đào sâu thích hợp :**

Đến cuối thập niên 1960 và đầu thập niên 1970, rất nhiều nhà nghiên cứu chỉ đơn thuần trình diễn những vấn đề cho người học trong khi thu thập và làm bảng kê những phản hồi (trả lời) của họ, mà không xem xét người học là một nhân tố trong toàn bộ hệ thống hướng dẫn. Nhiều nhà nghiên cứu đã phát triển những hệ thống làm thay đổi cách biểu diễn những “vật liệu” mới dựa trên thông tin của những phản hồi của người học. Trong khi nó không bao giờ đủ để biểu diễn thông tin trong một mẫu xác định trước cho tất cả người học nữa, những nhà thiết kế hệ thống còn phải lường trước tất cả những phản hồi có thể của người học. Những nhà lập trình phải biết cao hơn loại phản hồi nào của người học có thể xảy ra và quyết định nào mà hệ thống sẽ trình bày. Những hệ thống này là những hệ thống đầu tiên đã “mô hình hoá người học”, mặc dù chúng chỉ mô hình hoá hành vi của người học và không có bất kỳ cố gắng nào để mô hình hoá tình trạng tri thức của người học. Mặc dù những hệ thống tương đối đơn giản, bởi sự ràng buộc chính nó vào sự phát triển của những kĩ năng và khả năng nhớ lại mà chúng rất hiệu quả. Người học, những người sử dụng hệ thống này đã cải thiện những kĩ năng này và khả năng nhớ lại thực sự của mình. Lý thuyết về việc học cho rằng người học trước tiên cần phải học những kĩ năng và sự kiện cơ bản để chuẩn bị cho những kĩ năng tổng hợp cao hơn.

#### **1.3.4/ Sự khủng hoảng của Trí tuệ nhân tạo và Tâm lý học giáo dục :**

Trong suốt thời kỳ này, có một hy vọng lớn cho việc tiến triển nhanh trong trí tuệ nhân tạo nhờ sự tiến bộ cao trong sức mạnh tính toán ngày càng lớn của máy tính. Tuy nhiên nó sớm trở nên rõ ràng là những vấn đề trong trí tuệ nhân tạo khó hơn nhiều so với những thách thức tương đối dễ của việc xây dựng những máy tính nhanh hơn. Mặc dù vậy, những nhà nghiên cứu Trí tuệ nhân tạo vẫn còn lạc quan xuyên suốt thập niên 60, 70 và 80, ngay cả mục đích của những “máy tính biết suy nghĩ” luôn luôn dường như chỉ là “việc của 10 năm nữa”.

Khoảng thời gian này, tâm lý học giáo dục đang thắc mắc về những giả định của chủ nghĩa hành vi. Lý thuyết của Piaget về việc học và xây dựng bắt đầu được nói tới. Chomsky, cùng với Novell và những người khác đã giới thiệu những ý tưởng của tiến trình xử lý thông tin có hình ảnh, những ý tưởng mà gắn bó chặt chẽ với sự hứng thú của cộng đồng Trí tuệ nhân tạo trong ngôn ngữ học và xử lý ngôn ngữ tự nhiên. Tiến trình xử lý thông tin nổi lên như một mô hình bao quát xuyên suốt cuối thập niên 70 và đầu thập niên 80. Công việc này quan niệm sự nhận thức của con người như tập tiến trình xử lý “hộp đen”, hơn là những phản hồi đơn thuần từ bên ngoài. Sự chính xác của những mô hình xử lý thông tin đầy hứa hẹn khi được mô tả bởi những chương trình máy tính.

#### **1.3.5/ Trí tuệ nhân tạo và Hệ thống dạy học thông minh (AI – ITS) :**

Trong năm 1982, Sleeman và Brown đã xem xét lại thực trạng của những hệ thống hướng dẫn hỗ trợ dựa máy tính và lần đầu tiên tạo ra cụm từ “Intelligent Tutoring Systems (ITS)”- Những hệ thống dạy học thông minh, để miêu tả những hệ thống suy luận này và phân biệt chúng với những hệ thống CAI trước đó. Giả định không tưởng minh cho rằng người học bây giờ đã tập trung vào việc “học bằng cách làm”. Họ phân biệt những ITS tồn tại thành các loại như : những hệ thống theo dõi giải quyết vấn đề (1), những hệ thống huấn luyện (2), những hệ thống hướng dẫn thí

nghiệm (3), những hệ thống tư vấn (4). Điểm quan trọng của những hệ thống này thì còn là những nền tảng nghiên cứu cho việc tinh chế lý thuyết Trí tuệ nhân tạo, nhưng bây giờ các nhà nghiên cứu đang suy nghĩ về việc biểu diễn tri thức người học trong những hệ thống này. Ở đây, người ta đã tìm thấy đầu tiên việc sử dụng cụm từ “Mô hình người học” để miêu tả một sự biểu diễn trừu tượng về người học trong chương trình máy tính. Họ phân loại những mô hình người học như sau :

+ Mô hình che phủ : một mô hình tri thức người học như một tập con của một tri thức chuyên gia.

+ Mô hình khác biệt : tương tự như mô hình che phủ, mô hình này tập trung vào sự khác nhau giữa tri thức người học và một phần tri thức chuyên gia.

+ Mô hình trạng thái bất ổn : biểu diễn những khái niệm khó hiểu của người học như là những biến đổi của cấu trúc thủ tục của kỹ năng sửa đúng của chuyên gia.

Những cố gắng trước đây để mô hình hoá tri thức người học được dựa vào một mô hình “lỗi” (bugs). “Lỗi” là những sai sót của người học trong những kỹ năng rời rạc, chẳng hạn việc nhớ không đúng trong phép trừ. Burton đã thảo tỉ mỉ mô hình này trong hệ thống DEBUGGY. DEBUGGY định danh 130 “lỗi” được thiết kế để khắc phục những sai sót trong phép trừ. Thách thức là phải phân tích không gian vấn đề được trình bày bởi những câu trả lời của người học và xác định lỗi nào hay tập lỗi nào tốt nhất cho việc khắc phục những sai sót trong phép trừ.

Sleeman và Brown đề cập đến một vài phiên bản học tập có quan hệ tới những vấn đề liên quan đến việc tạo ra ITS. Họ thừa nhận rằng nhiều sự giao tiếp dạy học của con người không rõ ràng và nhấn mạnh đến kỳ vọng rằng ITS sẽ cung cấp “nơi gặp gỡ” cho những nhà lý thuyết về giáo dục để phát triển những lý thuyết rõ ràng chính xác hơn cho việc dạy và việc học. Giả định của họ là sự chính xác rõ ràng là có thể, là cần thiết cho việc thực thi những lý thuyết này trong phần mềm máy tính. Họ cũng thảo luận về nhu cầu phải xây dựng những môi trường để khuyến khích việc học cộng tác,

trong khi thừa nhận rằng những nhà nghiên cứu (tại thời điểm đó) biết rất ít về việc làm cách nào để sự cộng tác có thể xảy ra trong việc thiết lập việc học tự nhiên.

### 1.3.6/ Vai trò của khoa học nhận thức :

Trong suốt thập niên 1980, những nhà khoa học máy tính đang chuyên biệt hoá trong Trí tuệ nhân tạo tiếp tục tập trung vào những vấn đề của ngôn ngữ tự nhiên, mô hình người học và sự suy diễn. Tuy nhiên, lĩnh vực này cũng thu hút những nhà nghiên cứu khác, mà nổi bật là John Anderson. Anderson làm việc trong lĩnh vực khoa học nhận thức, phát triển lý thuyết nhận thức “Sự kiểm soát thích nghi của suy nghĩ” (ACT\* - Adaptive Control of Thought). Bảng 1.1 tóm tắt những nguyên tắc ACT\* và sự thực thi của chúng cho ITS.

**Bảng 1.1 : Những giả định ACT\* và những nguyên tắc liên quan cho việc dạy được thực thi trên máy tính**

Những giả thuyết ACT*	Những nguyên tắc dạy học tương ứng
Hành vi giải quyết vấn đề là mục đích hướng tới.	Giao tiếp với cấu trúc mục đích nằm bên dưới tác vụ giải quyết vấn đề.
Tri thức thủ tục và khai báo được chia ra. Những đơn vị tri thức thủ tục là những quy tắc NẾU_THÌ (IF_THEN) được gọi cho sự sản sinh.	Biểu diễn tri thức người học như một tập của sự sản sinh.
Sự thực hiện khởi đầu của một tác vụ được hoàn thành bởi việc áp dụng những thủ tục tổng quát cho những cấu trúc khai báo	Cung cấp sự hướng dẫn trong ngữ cảnh giải quyết vấn đề để tri thức người học phát triển thông qua sự ước lượng thành công cho kỹ năng hướng dẫn



Sự nảy sinh tác vụ đặc biệt phát sinh bởi việc áp dụng những sự nảy sinh yếu hơn cho tri thức thủ tục. Những sự nảy sinh tác vụ đặc biệt này làm nền tảng cho sự thực hiện hiệu quả hơn.	Cung cấp phản hồi tức thì dựa vào những kết quả của sự cố gắng
Là kết quả của sự rèn luyện thêm, sự sản sinh có thể bị trói buộc với những sự sản sinh có tỉ lệ lớn hơn	Điều chỉnh kích thước từng bước hướng dẫn trong tiến trình học tập
Người học duy trì tình trạng hiện tại của vấn đề trong một bộ nhớ làm việc bị giới hạn	Tối thiểu hoá việc tải bộ nhớ làm việc

Mặc dù Anderson và đồng nghiệp đã tạo ra ACT\* như là một lý thuyết nhận thức, họ tin rằng nó đủ nghiêm ngặt để kiểm thử bởi sự thực thi những nguyên tắc trong phần mềm máy tính. Hai ví dụ phổ biến nhất là Geometry Tutor (Koedinger & Anderson, 1993) và LISPITS (Hệ thống dạy học LISP thông minh). Hệ thống LISPITS, một chương trình để dạy việc học lập trình ngôn ngữ LISP, được thiết kế để thực thi những nguyên tắc này trong ngữ cảnh của “Mô hình truy vết”. Những cố gắng của LISPITS để mô hình những bước cần thiết để viết một chương trình LISP. Chương trình sau đó so sánh với những bước thực sự mà người học tạo ra với mô hình này. Corbett và Anderson gọi tiến trình theo dõi và sửa chữa là “Truy vết tri thức”. Mục đích của họ là một mô hình tinh thông, nơi mà mỗi người học thông thạo 95% những quy tắc cho một tập các bài tập được đưa ra trước khi di chuyển sang phần kế tiếp. Corbett và Anderson nhận thấy rằng người học sử dụng LISPITS hoàn thành những bài tập thông thạo nhanh hơn là làm việc một mình, nhưng tất nhiên là không nhanh bằng học với những giáo viên là con người.

#### **1.4/ Những thách thức đối với các hệ thống ITS:**

Sự phát triển hệ thống ITS được bắt đầu từ đầu mà không dựa trên một nền tảng nào sẽ dẫn đến vài kết quả không mong đợi trong việc xây dựng hệ thống như: giá cả phát triển cao, việc dùng lại những phần mềm đã tồn tại rất ít, những yêu cầu cơ sở bị hạn chế, ít có sự tiêu chuẩn hoá giữa những hệ thống đã phát triển, khó khăn trong việc đánh giá hệ thống và chi phí duy trì hệ thống cao.

**Việc tái sử dụng** lại những phần mềm đã tồn tại là việc không phổ biến trong việc phát triển một hệ thống ITS. Những phần mềm phổ biến rất có giá trị, cả trong lĩnh vực thương mại (ví dụ: trình soạn thảo, những bảng tính, công cụ vẽ và cơ sở dữ liệu) và trong lĩnh vực nghiên cứu (lên kế hoạch, những công cụ xử lý ngôn ngữ tự nhiên). Nhưng thật không may, những công cụ và phần mềm phổ biến đã tồn tại không dễ dàng tích hợp trong hệ thống ITS. Điều này đã dẫn đến việc gia tăng thời gian phát triển của toàn bộ hệ thống. Tuy nhiên những phần mềm phổ biến sẽ giúp cho việc tạo những phiên bản đầu tiên nhanh hơn. Một số hệ thống đã được xây dựng bằng cách sử dụng phần mềm phổ biến như Microsoft Excel (Microsoft, 1997), SuperCalc (Hiệp hội máy tính, 1996) và HyperCard (Apple, 1997).

Những yêu cầu cơ sở có thể bị ngăn cản khi hệ thống ITS được thực thi như một chương trình lớn yêu cầu những tài nguyên quan trọng. Và hệ thống ITS cũng có thể cần những phần cứng đặc biệt hoặc có thể được thực thi trong những ngôn ngữ mà có thể giới hạn việc phân phối những yêu cầu cơ sở. Việc tái sử dụng lại những công cụ cũng cho phép sự phát triển của những siêu văn bản, siêu phương tiện, đa phương tiện và những ứng dụng CD-ROM, và những cụm từ chuyên môn kỹ thuật hiện hành. Những ứng dụng này có thể đưa ra những yêu cầu cơ sở không mong đợi mà không thêm vào bất cứ lợi ích quan trọng nào. Những tiếp cận kỹ thuật để xây dựng hệ thống dẫn đến việc không thể đo lường những khả năng của kỹ thuật mới.

Có rất ít những sự tiêu chuẩn hoá giữa những hệ thống đã tồn tại. Sự phát triển “Ad hoc” của hệ thống ITS khuyến khích sự tái sử dụng của những kỹ thuật tiêu biểu

cho tri thức mới, phương tiện thông tin mới, những thành tựu trong kiến trúc và những sơ đồ dòng điều khiển. Việc ít sử dụng những thành phần mềm trong hệ thống là phổ biến, đặc biệt việc chia sẻ giữa chiến lược giảng dạy và tài liệu giảng dạy vì không có tiêu chuẩn thống nhất chung. Những phương pháp thiết kế cho hệ thống ITS đã được định nghĩa trong nhiều năm qua. Tuy nhiên nhiều phương pháp chưa được ứng dụng dễ dàng khi xây dựng hệ thống.

Một kết quả của việc thiếu sự tiêu chuẩn hoá là sự phức tạp của việc đánh giá hệ thống và sự khó khăn của việc so sánh những hệ thống đã phát triển. Vì vậy, tiến trình của việc quyết định sự mạnh yếu của việc thực thi hệ thống là không được tính đến. Điều này đã dẫn đến việc đánh giá những yếu tố bên trong và bên ngoài là rất khó.

Chi phí duy trì hệ thống ITS có thể phát sinh khi càng nhiều hệ thống được phát triển, càng nhiều những phác thảo trong phòng thí nghiệm ít chú ý đến sự chu cấp. Vì vậy, sự bổ sung của hệ thống và những yêu cầu chức năng mới có thể buộc những phần lớn của hệ thống phải gỡ bỏ hoặc thực thi lại ngay từ đầu. Đối với mô hình về tri thức lĩnh vực, mô hình người học, chiến lược giảng dạy và sự phát triển giao diện không có bộ tiêu chuẩn nào cho việc xây dựng hệ thống ITS. Kết quả, chúng là những lĩnh vực phụ thuộc nhau. Điều này gây khó khăn cho hệ thống nếu không nói là không thể trong việc bổ sung những lĩnh vực mới hoặc thêm những tính năng mới cho những lĩnh vực đã tồn tại.

Tất cả những nhân tố này đã góp phần làm cho việc xây dựng hệ thống ITS trở nên khó khăn hơn. Một vấn đề khác liên quan đến những kỹ năng cần thiết và những yêu cầu cá nhân để xây dựng những hệ thống tương tự như vậy.

Đầu tiên, nếu đội ngũ phát triển đang xây dựng một hệ thống ITS, kể cả những chuyên gia trong lĩnh vực phải trở thành những thành viên không thể thiếu của đội ngũ phát triển hoặc người phát triển phải trở thành chuyên gia trong lĩnh vực.

Thứ hai, người giáo viên phải xây dựng những thói quen của họ đối với nhiều lĩnh vực khác nhau của hệ thống giảng dạy. Một lần nữa, phần lớn những chuyên gia

trong lĩnh vực không cần phải có những kỹ năng yêu cầu cho quá trình phát triển này. Tạo ra việc giảng dạy dựa trên máy tính không chỉ yêu cầu về tri thức trong lĩnh vực đó mà còn tri thức khoa học và lập trình. Có 3 lĩnh vực tri thức chính cần thiết cho sự phát triển của hệ thống ITS:

- Tri thức về lập trình :
  - + Mã hoá những thể hiện của tri thức.
  - + Thực thi hệ thống giảng dạy và những mô hình của nó.
  - + Thực thi giao diện người dùng.
- Tri thức về những nhận thức khoa học :
  - + Định nghĩa những kỹ thuật biểu diễn những tri thức.
  - + Định rõ những chiến lược giảng dạy.
  - + Phát triển mô hình người học và cách sử dụng cho họ.
- Tri thức về lĩnh vực :
  - + Định nghĩa mô hình lĩnh vực và mô hình nhiệm vụ.

Thật không may những chuyên gia lại không có nền tảng về lập trình và những nhận thức khoa học, người lập trình lại không có những nhận thức khoa học và những tri thức về lĩnh vực mình làm, còn những nhà khoa học không có những tri thức về lĩnh vực đó và kinh nghiệm lập trình.

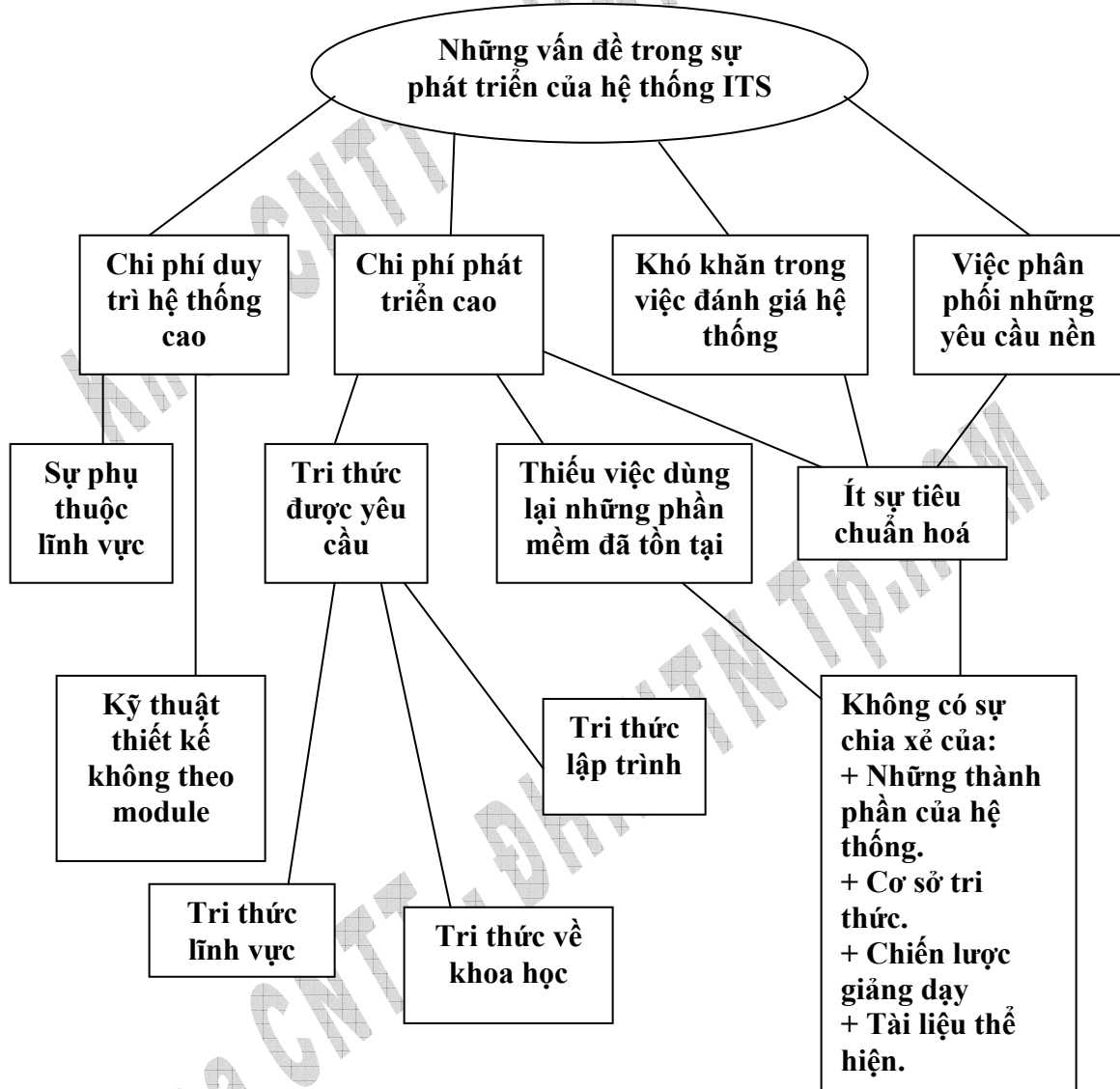
Một vấn đề khác liên quan đến chi phí cao của hệ thống ITS là việc tạo ra những sự phụ thuộc lĩnh vực. Hầu hết các hệ thống ITS có thể được nhóm thành 2 loại: sự độc lập các lĩnh vực và sự phụ thuộc các lĩnh vực.

- Hệ thống các lĩnh vực phụ thuộc nối các tri thức lĩnh vực, chiến lược giảng dạy và giao diện. Điều này gây khó khăn cho việc tái sử dụng, bổ sung và duy trì hệ thống. Hệ thống các lĩnh vực phụ thuộc hữu hiệu cho các lĩnh vực nhỏ và đã được định nghĩa tốt.

- Hệ thống các lĩnh vực độc lập giữ một vài thành phần của hệ thống độc lập khỏi tri thức lĩnh vực. Điều này cho phép hệ thống được sử dụng trong nhiều lĩnh vực

khác nhau. Tuy nhiên hệ thống các lĩnh vực độc lập vẫn còn giới hạn bởi một số môi trường nhất định, ví dụ: bộ mô phỏng, những lĩnh vực về toán học hoặc ngôn ngữ.

Một giải pháp cho những vấn đề này là cung cấp những công cụ và giải pháp để làm giảm chi phí trong sự phát triển của hệ thống ITS.



**H1.2: Những vấn đề trong việc phát triển hệ thống ITS**

### **1.5/ Sơ lược về kiến trúc của một hệ thống ITS :**

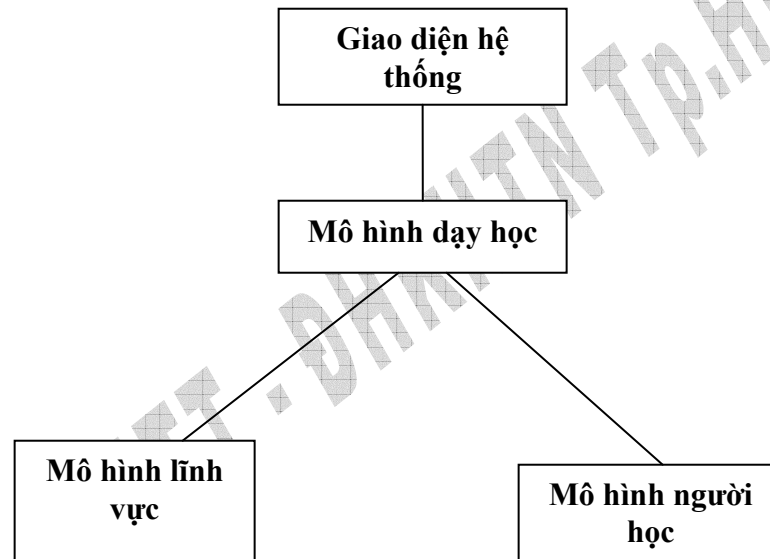
Những hệ thống dạy học hỗ trợ bởi máy tính thông thường thiếu chiều sâu về tri thức vì vậy hệ thống ITS đã nỗ lực để vượt qua những sự thiếu hụt này bằng cách sử dụng kỹ thuật trí tuệ nhân tạo.

Việc sử dụng kỹ thuật trí tuệ nhân tạo trong việc phát triển hệ thống ITS đã được mở rộng. Tuy nhiên có 3 kỹ thuật Trí tuệ nhân tạo đáng chú ý trong hệ thống giáo dục là lên kế hoạch, xử lý ngôn ngữ tự nhiên và kỹ thuật biểu diễn tri thức. Những kỹ thuật Trí tuệ nhân tạo có thể được dùng để cải tiến việc chẩn đoán những lỗi của người học, nhận ra những quan niệm không đúng của người học, những kiến trúc và kế hoạch của chiến lược giảng dạy.

Sự phát triển của những kỹ thuật biểu diễn tri thức trở nên quan trọng cho hệ thống ITS khi nhiều hệ thống yêu cầu việc sử dụng mô hình người học và mô hình lĩnh vực.

Có nhiều kiến trúc khác nhau tùy thuộc vào các hệ thống ITS khác nhau. Nhưng theo một số chuyên gia thì có 4 thành phần phổ biến trong hầu hết các hệ thống ITS là: sự biểu diễn tri thức của lĩnh vực ( Mô hình lĩnh vực), chiến lược giảng dạy sử dụng để giải quyết những vấn đề đó (Mô hình dạy học), sự biểu diễn của tình trạng hiện hành của người học qua những vấn đề đó (mô hình người học) và giao diện cho sự tương tác giữa người sử dụng và hệ thống (giao diện hệ thống).

Hình sau biểu diễn mô hình hệ thống ITS phổ biến:



### H1.3 : Mô hình ITS thông thường

Sự đa dạng trong mô hình này có thể được tìm thấy trong nhiều phiên bản. Tuy nhiên, cái lõi của từng cấp độ đều có kiến trúc chung cơ bản.

#### 1.5.1/ Mô hình lĩnh vực (Tri thức về lĩnh vực) :

Tri thức về lĩnh vực là những thông tin phụ thuộc vào lĩnh vực mà nó được dùng trong suốt quá trình giảng dạy.

Có nhiều nỗ lực để định nghĩa kỹ thuật biểu diễn tri thức đã được làm trong quá trình phát triển của hệ thống ITS trong nhiều năm qua. Những kỹ thuật đã được phát triển để cho phép những tri thức lĩnh vực được lưu trữ và truy cập có hiệu quả. Chẳng hạn trong hệ thống DIT do Chiu-Chen Hsied, Tzong-Han Tsai, David Wible, Wen-Lian Hsu đã biểu diễn tri thức bằng cách sử dụng InfoMap. Nó tương tự với cấu trúc cây được hiển thị trong Hình 1.4. Mỗi nút thuộc hai loại chính là : nút khái niệm hoặc nút chức năng.



H1.4 : Biểu diễn tri thức trên InfoMap

Gần đây với sự phát triển nhanh của XML (Extensible Markup Language) cùng với nhiều lợi ích của nó, nhiều hệ thống ITS đã sử dụng XML để biểu diễn tri thức trong Mô hình lĩnh vực. Hình sau minh họa cho việc sử dụng này ( được sử dụng trong hệ thống của Steven Linton).



```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xs:element name="ER">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element ref="entity"/>
        <xs:element ref="relation"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>

  <xs:element name="entity">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="attribute" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:string" use="required"/>
    </xs:complexType>
    <xs:key name="entity_key">
      <xs:selector xpath="."/>
      <xs:field xpath="name"/>
    </xs:key>
  </xs:element>

  <xs:element name="relation">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="attribute" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:string" use="optional"/>
      <xs:attribute name="from" type="xs:string" use="required"/>
      <xs:attribute name="to" type="xs:string" use="required"/>
      <xs:attribute name="card" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="\d+, (\d+ | unbounded) -\d+, (\d+ |
unbounded)"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
    <xs:keyref name="from_ref" refer="entity_key">
      <xs:selector xpath="."/>
      <xs:field xpath="from"/>
    </xs:keyref>
    <xs:keyref name="to_ref" refer="entity_key">

```

```

<xs:selector xpath="."/>
<xs:field xpath="to"/>
</xs:keyref>
</xs:element>

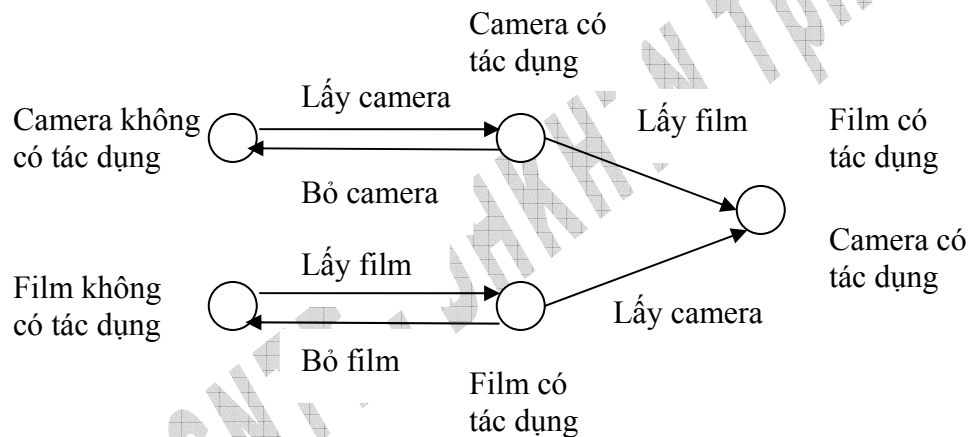
<xs:element name="attribute">
<xs:complexType>
<xs:attribute name="name" type="xs:string" use="required"/>
<xs:attribute name="type" type="xs:string" use="required"/>
</xs:complexType>
</xs:element>
</xs:schema>

```

### H1.5 : Một định nghĩa XML Schema cho một thực thể trong Mô hình lĩnh vực

Sự lựa chọn kỹ thuật biểu diễn tri thức cho một lĩnh vực nào đó là một quyết định quan trọng trong việc phát triển một hệ thống ITS. Một vài kỹ thuật biểu diễn tri thức đã “mượn” từ Trí tuệ nhân tạo, việc nghiên cứu hệ chuyên gia (mạng ngữ nghĩa, các bộ luật). Một trong những kỹ thuật được dùng nhiều nhất là mạng các công việc chuyển tiếp dựa trên trạng thái.

#### Ví dụ:



Trong ví dụ trên thì các tri thức lĩnh vực được lưu là các trạng thái (camera không có tác dụng, camera có tác dụng, film không có tác dụng, film có tác dụng) và

các hoạt động (lấy film, lấy camera, bỏ film, bỏ camera) và các tri thức đó được lưu dưới dạng sơ đồ trạng thái.

### 1.5.2/ Mô hình người học :

Mô hình người học trong một hệ thống ITS lý tưởng là có cái nhìn chính xác về kiến thức của người đang sử dụng hệ thống. Mô hình người học bao gồm sự tin tưởng của hệ thống vào kiến thức của người học. Nó không tương tác với người học mà nó đại diện cho người học. Từ mô hình này, hệ thống có thể xác định những vùng tri thức chưa chính xác hoặc còn thiếu và nhắm những lĩnh vực này vào trong tài liệu giảng dạy. Mô hình người học phải kết hợp khả năng của người học cùng với yêu cầu về lĩnh vực mà người đó chọn.

Ba vấn đề xác đáng để nghiên cứu trong mô hình người học và cái gì cần phải được xem xét:

- Thông tin nào của người học cần phải được đưa ra?
- Thông tin được biểu diễn như thế nào?
- Mô hình người học được xây dựng như thế nào?

Thông thường những thông tin chứa trong mô hình người học là kiến thức mà người học cần phải học về lĩnh vực hiện hành. Điều này sẽ bao gồm cả những cái gì mà người học đã học rồi và những cái gì mà người học chưa học. Những thông tin khác có thể có trong mô hình người học là :

- Kiểu tương tác người học với hệ thống.
- Mức độ khả năng của người học.
- Xem xét khả năng nhớ của người học.
- Mục đích muốn đạt được của người học cho vấn đề.

Có 3 kỹ thuật cho việc biểu diễn thông tin được chứng minh là có hiệu quả trong việc xây dựng mô hình người học. Đó là: mô hình che phủ, mô hình khác biệt và mô hình trạng thái bất ổn.

Một trong những kỹ thuật mô hình hoá người học đơn giản nhất được sử dụng trong các hệ thống ITS là sự mô hình che phủ. Nó xem xét mô hình người học như là một tập con của mô hình tri thức. Mô hình người học có thể được sử dụng để định danh những phần thiếu sót khi so sánh với mô hình tri thức. Vì vậy, tiến trình dạy học có thể tập trung vào phần này cho đến khi mô hình người học và mô hình tri thức là tương đương.

Tuy nhiên, sự mô hình hoá che phủ có một vài thiếu sót quan trọng. Mô hình bị giới hạn cho tri thức được biểu diễn trong mô hình lĩnh vực. Vì vậy nếu người học cố gắng thay đổi (chiến lược hợp lý tiềm năng) thì đầu tiên hệ thống có lẽ không nhận ra. Thứ hai là, nó có lẽ ở ngoài phạm vi của tri thức hệ thống. Do đó hệ thống không có khả năng giải thích tại sao câu trả lời là sai. Nếu mô hình lĩnh vực chỉ là một mô hình thì nó sẽ không hoàn chỉnh. Do đó, mô hình người học dựa trên tri thức không hoàn chỉnh có thể giới hạn khả năng giải thích của hệ thống ITS.

Một sự biến đổi cho mô hình che phủ là mô hình khác biệt. Ở đây, mô hình người học được so sánh với mô hình mô tả những gì mà một chuyên gia sẽ làm trong tình huống giống nhau. Tri thức lĩnh vực được chia thành 2 phần : tri thức mà người học nên có và tri thức mà người học không mong muốn có. Mô hình khác biệt một lần nữa xem mô hình người học là một tập con của mô hình tri thức và vì vậy cũng bị ràng buộc bởi sự giới hạn không hoàn toàn tương tự nhau như mô hình che phủ.

Một kỹ thuật mô hình hoá người học thứ ba là sự mô hình “trạng thái bất ổn”. Tương tự với những mô hình trước đó, một liên kết đóng được giữ giữa mô hình người học và mô hình chuyên gia trong tri thức lĩnh vực. Thêm vào đó, kỹ thuật này cho phép mô hình người học được mở rộng trên mô hình chuyên gia. Vì vậy, mô hình người học và mô hình chuyên gia cơ bản giống nhau với những sự khác biệt nhỏ trong một vài

phần của tri thức lĩnh vực. Việc sử dụng thư viện lỗi là một kỹ thuật phổ biến dùng trong mô hình này.

Trong hệ thống CIRCSIM-Tutor, Yujian Zhou và Martha W.Evens (1999) xây dựng Mô hình người học thành 4 thành phần chính :

- (i) Mô hình hiệu suất thực hiện : đánh giá khả năng của người học theo các mức.
- (ii) Thông tin giải quyết vấn đề của người học : lưu lại những câu trả lời của người học để quyết định chiến lược phản hồi.
- (iii) Tài liệu giải quyết vấn đề của người học: mô hình này lưu lại số lỗi mà người học tạo ra khi giải quyết vấn đề.
- (iv) Mô hình thông tin dạy học : mô hình này bao gồm cả thông tin việc lên kế hoạch và thuyết trình.

Mặc dù mô hình người học được sử dụng giống như nền tảng của việc dạy học trong nhiều hệ thống, một vài nhà nghiên cứu tin rằng việc hữu dụng của chúng bị giới hạn và mô hình người học phức tạp đó có thể không được xây dựng. Đối với những lĩnh vực giảng dạy mà phức tạp hơn những vấn đề bình thường trong số học đơn giản, việc định nghĩa mô hình người học và những lời giải thích cho những hành động của người học đã trở thành những vấn đề không hy vọng.

### **1.5.3/ Mô hình dạy học :**

Mô hình dạy học là nòng cốt của hệ thống ITS. Ở cấp độ nền tảng, nó có thể được dùng để tập hợp và định dạng thông tin từ thành phần tri thức lĩnh vực và hiển thị nó cho người học. Tuy nhiên mô hình dạy học có thể phức tạp hơn, cung cấp nhiều chiến lược dạy thích nghi với người học hiện hành, tổ chức việc cập nhật và duy trì mô hình người học. Mô hình dạy học thường có nỗ lực tự động hoá vai trò của người dạy.

Phản hồi (feedback) là một trong những tính năng quan trọng của Mô hình dạy học, góp phần tạo nên sự thông minh của hệ thống. Hệ thống phải có khả năng đưa ra những hướng dẫn phản hồi đúng lúc trong quá trình học của người học để giúp chúng

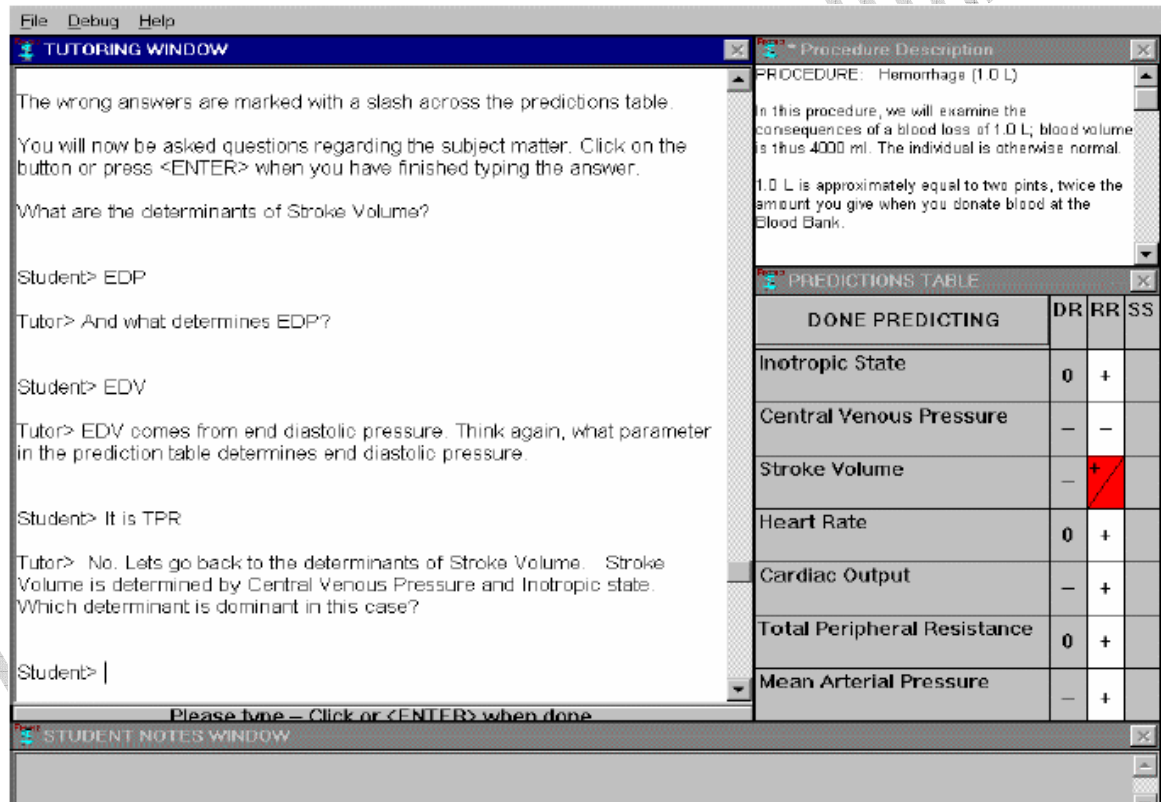
có thể giải quyết vấn đề. Phản hồi có thể dùng văn bản hoặc hình ảnh, âm thanh để tăng tính hiệu quả và thu hút người học. Những hệ thống được đánh giá tốt khi sự phản hồi hướng dẫn của hệ thống hoạt động “linh hoạt và thông minh”.

Việc sử dụng bộ mô phỏng trong hệ thống ITS là một hữu hiệu đặc biệt cho Mô hình dạy học khi một thiết bị được đưa ra.. Mô hình dạy học mô phỏng hoạt động của một vài thiết bị và xử lý tương tác của người học với những mô hình thiết bị đó. Người học sẽ có được những phản hồi ngay lập tức từ những hoạt động thực hiện trong môi trường mô phỏng. Tuy nhiên lợi ích mà hệ thống ITS có thể mang lại cho quá trình giảng dạy phụ thuộc vào mức độ mà người học tương tác với hệ thống, điều đó phụ thuộc vào giao diện của hệ thống.

#### **1.5.4/ Giao diện của hệ thống :**

Giao diện của hệ thống có thể nói là thành phần chủ chốt trong việc thành công của một hệ thống ITS. Không có giao diện hợp lý người sử dụng không thể có được những lợi ích của hệ thống. Với một hệ thống máy tính nói chung giao diện người dùng đóng vai trò quan trọng trong sự thành công của hệ thống. Nếu người sử dụng không thoả mãn với giao diện, những hiệu quả của hệ thống sẽ giảm bớt và có thể bị huỷ bỏ. Giao diện người dùng cũng được xem quan trọng với những hệ thống hỗ trợ dạy học dựa máy tính bởi vì giao diện người dùng không nên làm cản trở quá trình học tập. Điều này có thể trong trường hợp hệ thống bị hỗn lộn bởi một loại giao diện bị tái sử dụng cho nhiều lĩnh vực khác nhau.

Thông thường, các hệ thống trước đây đều dựa vào những kĩ thuật xử lý ngôn ngữ tự nhiên để tạo những giao diện văn bản thân thiện người dùng. Chẳng hạn, hệ thống CIRCSIM-Tutor (Yujian Zhou và Martha W.Evens,1999). Hình 1.6 là màn hình chính của hệ thống. Hệ thống sẽ trình bày cho người học bảng mô tả sự thay đổi sinh lý trong cửa sổ ở góc trên phải. Hệ thống sau đó sẽ hỏi người học dự đoán tác động của sự thay đổi này lên 7 thông số sinh lý quan trọng (màn hình bên phải). Hệ thống sẽ “đổi thoại” với người học để sửa sai những lỗi trong các dự đoán của người học.



### H1.6 : Màn hình chính của hệ thống CIRC-SIM-Tutor

Ngày nay, kỹ thuật đa phương tiện đã được dùng để nâng cao giao diện người dùng trong kỹ thuật ITS. Với ngày càng nhiều công cụ hình ảnh – âm thanh mới, nhiều tiện lợi, giao diện cũng ngày càng thân thiện, tinh vi, dễ sử dụng cũng như xây dựng. Những giao diện mới dạng này sẽ thu hút nhiều hơn người học, làm cho họ không bị nhàm chán với các hoạt động thay đổi khác nhau. Và cuối cùng, thành phần giao diện của hệ thống ITS phải cung cấp sự liên kết thích hợp giữa người sử dụng và Mô hình dạy học.

### **1.6/ Kết luận :**

Trong thập niên vừa qua, hệ thống ITS đã “di chuyển” ra ngoài các phòng thí nghiệm và được sử dụng trong các lớp học hay “phòng huấn luyện” nơi mà một vài hệ thống đã tỏ ra có hiệu quả. Những hệ thống dạy học thông minh đang trở nên phổ biến và cho thấy một hiệu quả tăng cao, mặt khác chúng ta gặp nhiều khó khăn và công sức cũng như tiền bạc để xây dựng những hệ thống như vậy. Những hệ thống soạn thảo có tính khả thi cao hơn cho sự hướng dẫn hỗ trợ bởi máy tính và huấn luyện dựa trên phương tiện đa truyền thông, nhưng hệ thống soạn thảo này thiếu sự tinh vi đòi hỏi để xây dựng những hệ thống dạy học thông minh (ITS). Những hệ thống soạn thảo đa truyền thông thương mại có những công cụ thiết kế hướng dẫn để tạo ra những giao diện tương tác hình ảnh “hấp dẫn và lôi cuốn”, nhưng khả năng sư phạm và sự biểu diễn nội dung khá hạn chế. Những nhà nghiên cứu đang đầu tư nhiều vào những công cụ soạn thảo ITS từ khi những hệ thống ITS bắt đầu được nghiên cứu, và rất nhiều công cụ soạn thảo đã được xây dựng. Nội dung này sẽ được thảo luận kĩ hơn trong chương 2 - Những công cụ soạn thảo ITS (ITSAT).



## **Chương 2 : Những công cụ soạn thảo ITS ( ITS Authoring Tools – ITSATs)**

### **2.1/ Giới thiệu :**

Việc xây dựng một ITS là một nhiệm vụ không tầm thường. Một cách để làm đơn giản hoá công việc này là tạo ra các công cụ để hỗ trợ cho việc phát triển ITS. Việc sử dụng những công cụ như vậy có nghĩa là mỗi việc phát triển ITS không phải xây dựng ngay từ đầu. Điều này thường đạt được bằng cách tái sử dụng những thành phần chung giữa các dự án đó.

Chương này nghiên cứu về bản chất của ITSAT và mối quan hệ của chúng với tiến trình soạn thảo ITS. Có một vài vấn đề về sự tồn tại của những công cụ soạn thảo ITS bao gồm sự lựa chọn những phương pháp cho việc biểu diễn tri thức và thiếu việc tái sử dụng các kiến trúc ITS. Những vấn đề này được mô tả trong chương này cũng như sự giống nhau giữa ITSAT và ESS ( Expert System Shells – Vỏ hệ chuyên gia).

Những yêu cầu của một ITSAT là một công việc quan trọng, đặc biệt là một công cụ tổng quát hay một công cụ đặc biệt hoá thích hợp hơn. Những công cụ tổng quát hoá có thể cung cấp một môi trường cho việc định nghĩa nhiều lĩnh vực khác nhau nhưng có lẽ chỉ cung cấp một cơ sở dạy học bình thường. Những công cụ đặc biệt hoá có thể cho phép tri thức chuyên sâu cho việc dạy học nhưng lại đòi hỏi chi phí lớn.

Trong phần sau chương này sẽ mô tả những mục đích chính của việc xây dựng ITSAT, cũng như những yêu cầu của một ITSAT. Sau đó sẽ phân loại những công cụ soạn thảo dựa trên những công cụ đã phát triển. Và cuối chương sẽ thảo luận những phần cần được soạn thảo của một ITS sử dụng ITSAT, và những phương pháp soạn thảo căn bản.

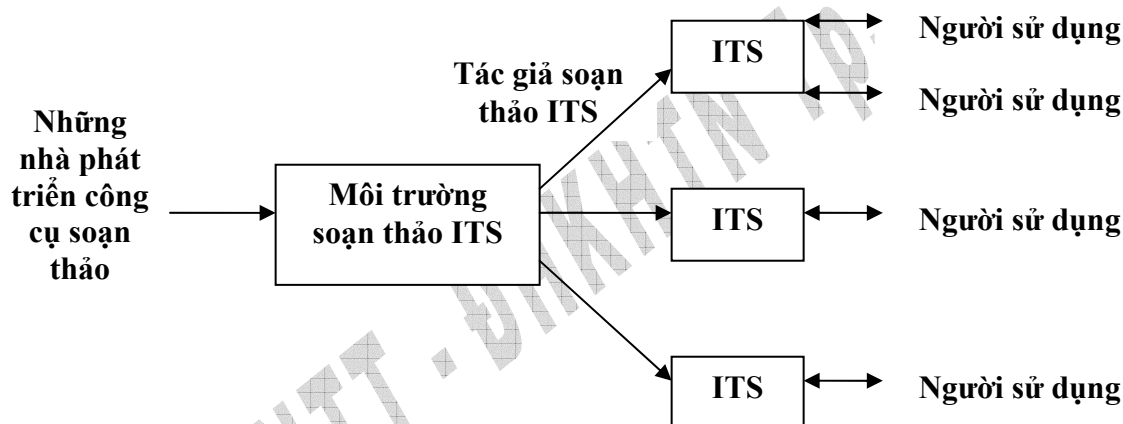
## **2.2/ Công cụ soạn thảo ITS - Mục đích :**

### **2.2.1/ Công cụ soạn thảo ITS :**

Trong những năm gần đây, có sự quan tâm đáng kể đến việc sử dụng những công cụ phần mềm đặc biệt để hỗ trợ trong tiến trình phát triển ITS. Mối quan tâm này đã được khuyến khích bởi cả sự thật là việc xây dựng ITS là công việc tốn nhiều thời gian và tiền bạc, và có rất ít tiêu chuẩn giữa những hệ thống đã được phát triển hay ngay cả những thành phần của các hệ thống. Hai thành phần rõ ràng cho sự tự động của việc xây dựng giao diện người dùng và sự đạt được tri thức. ITS cũng có thể đã được xây dựng hiệu quả hơn nếu những thành phần được tái sử dụng giữa những lần phát triển, chẳng hạn, cơ sở tri thức và chiến lược dạy học.

Một mục đích của những môi trường soạn thảo là để cung cấp một công cụ mà một chuyên gia biểu diễn tri thức không biết lập trình có thể dễ dàng sử dụng để tạo ra ITS theo yêu cầu. Cuối cùng, sự biểu diễn ở mức cao được sử dụng được che dấu những kĩ thuật bên dưới để người sử dụng không cần phải là một kĩ sư tri thức tài giỏi, một nhà phát triển giao diện, và cũng không cần phải là một chuyên gia về lĩnh vực ứng dụng cũng có thể tạo ra một ITS theo yêu cầu .

Có 3 tác nhân khác biệt trong lĩnh vực của việc sử dụng những công cụ soạn thảo để hỗ trợ cho việc phát triển ITS: những nhà phát triển công cụ soạn thảo; những tác giả soạn thảo ITS và những người sử dụng ITS. Những nhà phát triển công cụ soạn thảo có liên quan đến việc xây dựng môi trường soạn thảo ITS. Những người sử dụng của môi trường này là những tác giả soạn thảo ITS mà họ đã sử dụng môi trường này để xây dựng hệ thống tùy biến của họ. Tiếp theo nó được sử dụng bởi người học bằng cách tương tác với ITS. H2.1 trình bày môi trường phát triển ITS và những tác nhân tương tác với nó.



## H2.1 Tổng quan về môi trường phát triển ITS

Soạn thảo một ITS là một nhiệm vụ phức tạp. Sự phát triển của một công cụ soạn thảo thậm chí là khó khăn hơn khi nó ám chỉ đến những vấn đề ITS được định danh, cũng như những phiên bản riêng của nó. Vì vậy những vấn đề của việc phát triển một ITSAT thực sự là một siêu tập các phiên bản có liên quan khi xây dựng một ITS. Không chỉ chúng phải được xem xét mà những vấn đề này còn cần được hiển thị cho các tác giả soạn thảo ITS. Những phiên bản đặc thù là về những yêu cầu soạn thảo, mức kiểm soát hệ thống, việc sử dụng những mô hình nhận thức và làm cách nào để đạt được cơ sở tri thức và việc tái sử dụng được khuyến khích và hiển thị cho các tác giả soạn thảo ITS.

Có hai vấn đề chính cần được nêu trong việc soạn thảo ITS : đầu tiên là, ITS là một hệ thống phức tạp và đòi hỏi sự phát triển của sự biểu diễn tri thức đặc biệt cho lĩnh vực, sự hướng dẫn và tri thức người học và giao diện người dùng đặc biệt cho từng lĩnh vực. Thứ nhì là, có rất ít việc tái sử dụng những kiến trúc của ITS hiện tại thông qua những ứng dụng. Khả năng tái sử dụng được định nghĩa như là một tập của khả năng có thể phục hồi và khả năng thích ứng. Vì vậy, tái sử dụng có thể được xem một

cách tổng quát như một cách dễ dàng mà những thành phần ITS có thể được khôi phục và định rõ cho những môi trường mới.

Trong nhiều năm, có rất nhiều hệ thống thương mại đã có để hỗ trợ trong việc soạn thảo phần mềm dạy học. Nó thực sự hữu ích nếu những sản phẩm như vậy có thể được sử dụng trong sự phát triển của ITS, cũng như chúng có rất nhiều tính năng có ích. Không may là nhiều những hệ thống thương mại này không có những thành phần thông minh và cần có những “hành vi thích nghi” để được lập trình vào trong chúng. Điều này đòi hỏi không chỉ tri thức lập trình đặc biệt mà còn có sự hiểu biết chi tiết về các kỹ thuật trí tuệ nhân tạo. Vì vậy, hầu hết những hệ thống cho kết quả như là những hệ thống huấn luyện dựa trên máy tính (Computer-Based Training - CBT) hay những hệ thống hướng dẫn hỗ trợ bởi máy tính (Computer-Aided Instruction – CAI) bị giới hạn.

Cần hiểu rõ sự khác biệt giữa CBT và ITS ở đây là một yêu cầu cho một môi trường phát triển thông minh, bao gồm những phương pháp khởi đầu được hỗ trợ bởi công cụ phần mềm mà làm cho dễ dàng cả việc thiết kế của sự biểu diễn tri thức và sự phiên dịch những mô tả như vậy suốt các phiên dạy học. Điều này là một khó khăn chủ yếu của việc phát triển ITS. Tri thức cho một ứng dụng phải được đòi hỏi và lưu trữ trong một định dạng mà có thể sử dụng một cách có hiệu quả bởi những phương pháp dạy học. Vấn đề chính này là khởi đầu mà những tác giả soạn thảo có thể truy cập những phương pháp và chiến lược chính thức mà không cần phải trở thành những lập trình viên máy tính tài giỏi hay những kỹ sư về tri thức.

Thực sự khó để định nghĩa rõ ràng của những thành phần nào nên có trong một ITSAT. Vấn đề cung cấp một định nghĩa tổng quát cho một ITSAT là tương tự với vấn đề cố gắng định nghĩa một ITS tổng quát. Lĩnh vực và dãy ứng dụng của ITS cần nhắm tới cần được xem xét trong định nghĩa ITSAT của nó. Ta không thể xác định tốt những thành phần nào của một hệ thống ITS nên có thể được soạn thảo. Do đó cơ sở tri thức

lĩnh vực là bắt buộc cơ bản với những khả năng có thể của việc soạn thảo những chiến lược dạy học và những quy tắc đối thoại giữa người dạy và người học.

Ngay cả nếu những thành phần xác định được xem xét như những “ứng viên” quan trọng cho việc hình thành các thành phần của một ITSAT thì có bao nhiêu trong số chúng sẽ là những thành phần mặc định và có bao nhiêu sẽ là tùy biến. Có nên cho Lĩnh vực độc lập với Engine dạy học hay không? Và có nên cho những phương pháp dạy học có thể được định nghĩa lại hoàn toàn bởi tác giả soạn thảo hay không? Cũng vậy, làm cách nào để những giao diện người dùng thân thiện có thể được định nghĩa cho những hệ thống tổng quát?

Đối với một ITS tổng quát, hướng tiếp cận có thể nhận biết nhất là một mô hình ITS chuẩn, bao gồm **Tri thức lĩnh vực, một engine dạy học, những phương pháp dạy học và một giao diện người dùng**, cũng như một nền tảng cho những thành phần được soạn thảo. Sau đó, trong phạm vi của môi trường lĩnh vực cần nhắm tới như mong muốn, nó có thể được xác định những thành phần nào nên tùy biến.

Những loại của những hệ thống xây dựng hệ thống dạy học có sẵn là cực kỳ khác nhau. Trong những năm gần đây, có sự tăng đáng kể về số lượng các công cụ thương mại dùng cho việc xây dựng những hệ thống dạy học. Hệ thống này là những công cụ xây dựng CAI khởi đầu dùng để hỗ trợ tác giả soạn thảo trong các định nghĩa của phần mềm dạy học. Ta có thể xem hai lớp khác nhau của những hệ thống soạn thảo. Đầu tiên là, những hệ thống thương mại, Authorware, ToolBook và SmartText và thứ hai là, những công cụ nghiên cứu IDE, KAFiS, COCA và GTE.

Những hệ thống thương mại cho phép thiết kế hướng dẫn được khai báo. Những công cụ này cho phép một “lối vào” dễ dàng và hiển thị thông tin dưới dạng văn bản, đồ họa, phim và âm thanh. Sự giúp đỡ với bài học đã được lên kế hoạch và sự cung cấp những thành phần giao diện được định nghĩa trước làm dễ dàng nhiệm vụ hướng dẫn dạy học của máy tính. Những công cụ này cho phép một tác giả soạn thảo,

chẳng hạn một nhà thiết kế hướng dẫn, để tạo ra phần mềm dạy học đơn giản một cách nhanh chóng. Hai thuận lợi chủ yếu của những công cụ này là chúng bao gồm số lượng lớn tài nguyên cho việc định nghĩa và điều khiển tác động trực tiếp những đối tượng giao diện và việc dễ dàng sử dụng.

Tuy nhiên, những công cụ thương mại không cung cấp bất kỳ “Module thông minh” nào cho những hệ thống mà họ xây dựng. Tất cả những “hành vi thích nghi” được lập trình trước bởi người thiết kế hệ thống. Điều này dẫn đến những sự tương tác dạy học bị động phần lớn vào hệ thống, điều này bởi vì thiếu một ngôn ngữ kịch bản có ý nghĩa để mã hoá những hành vi thông minh bên trong những công cụ này.

### **2.2.2/ Mục đích :**

Xây dựng một hệ thống ITSAT hoàn chỉnh hay có thể chấp nhận được là vô cùng khó khăn và tốn kém, nhiều khi là không thể. Nhưng những nhà phát triển vẫn nghiên cứu và đi theo hướng này vì những lợi ích mà nó sẽ mang lại sau này. Sau đây là một vài mục đích của những công cụ soạn thảo :

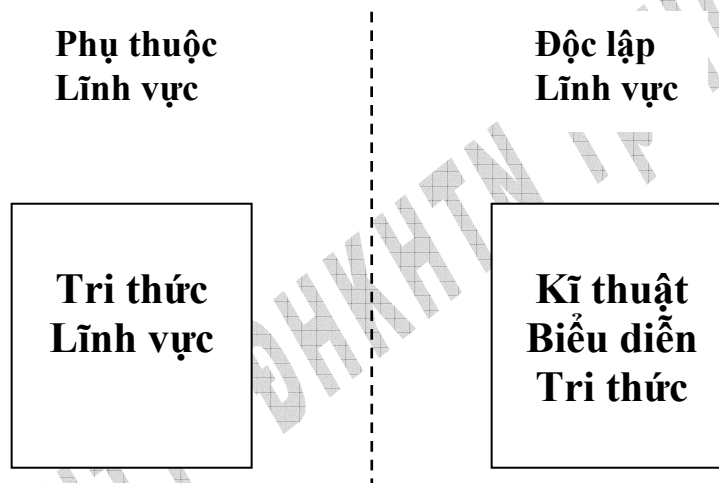
- + Giảm nỗ lực (thời gian, chi phí, và/hay những tài nguyên khác) cho việc tạo ra những hệ thống dạy học thông minh.
- + Giảm ngưỡng kỹ năng cần cho việc xây dựng những hệ thống dạy học thông minh (chẳng hạn cho phép nhiều người tham gia vào tiến trình thiết kế).
- + Giúp cho người thiết kế / tác giả soạn thảo làm khỏp hay tổ chức tri thức lĩnh vực .
- + Hỗ trợ (chẳng hạn cấu trúc, đề nghị , hay bắt buộc) những nguyên tắc thiết kế tốt (trong dạy học , giao diện người dùng ...)
- + Cho phép tạo nhanh các mẫu đầu tiên của những thiết kế hệ thống dạy học thông minh (chẳng hạn cho phép chu kỳ thiết kế / đánh giá nhanh của bản mẫu phần mềm đầu tiên).

### **2.3/ So sánh giữa ESS (Expert System Shells) và ITSAT :**

Ở một vài khía cạnh, những công cụ soạn thảo ITS có thể được xem xét so sánh với ESSs được sử dụng để trợ giúp sự phát triển của các hệ chuyên gia. Tương tự với sự phát triển của ESS, sự phát triển của ITSAT được điều hướng theo nhu cầu của việc giảm chi phí phát triển hệ thống. ITSAT và ESS được dự định để cho phép những người không biết lập trình hưởng lợi từ nỗ lực của những người khác mà đã giải quyết vấn đề tương tự vấn đề của riêng họ. Một khung nền cơ bản được cung cấp trong cả 2 hệ thống, một engine tham chiếu cho một hệ chuyên gia và một liên kết giữa engine dạy học hay những chiến lược dạy học cho một ITS. Tri thức lĩnh vực liên quan cho lĩnh vực hiện tại, sau đó có thể được thêm vào bên trong hệ thống bởi tác giả soạn thảo.

Không phải tất cả ESS đều phù hợp với tất cả các nhiệm vụ. Điều này cũng đúng cho ITSAT. Với cả những loại công cụ phát triển cho một thỏa hiệp giữa tính tổng quát của công cụ và sức mạnh của hệ thống được xây dựng. Những nhà xây dựng công cụ một cách tự nhiên muốn phát triển những công cụ có mục đích tổng quát mà có thể ứng dụng rộng rãi cho nhiều vấn đề.

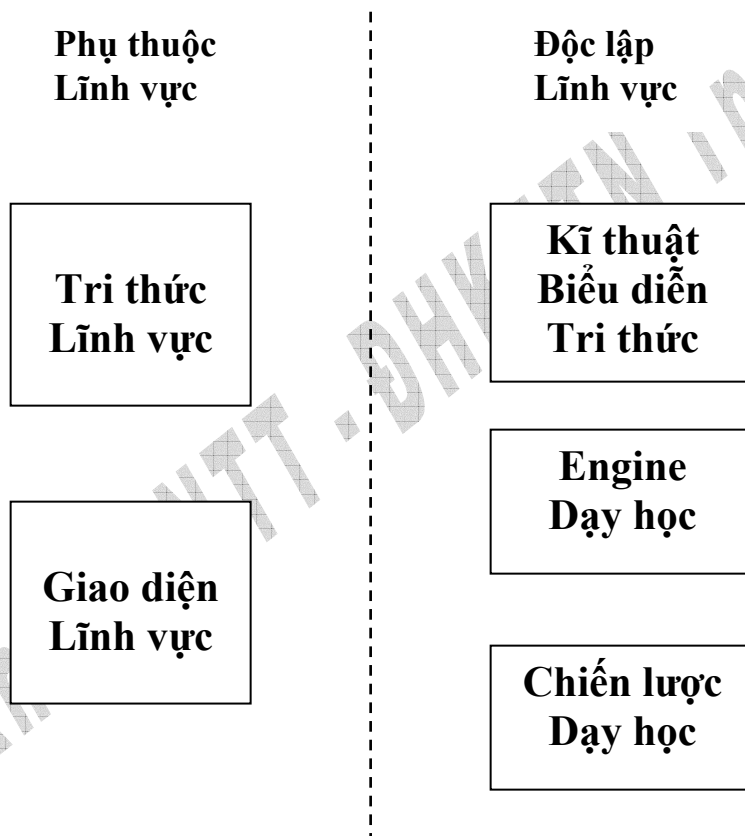
Tương tự như ESS, một mức nào đó của tính độc lập lĩnh vực ITS có thể đạt được bằng cách giữ tất cả tri thức phụ thuộc lĩnh vực trong cơ sở tri thức và sử dụng những phương pháp độc lập lĩnh vực cho việc biểu diễn tri thức trong cơ sở tri thức (Hình 2.2).



### H2.2 Sự phân chia kĩ thuật biểu diễn và tri thức lĩnh vực

Sự phân chia của cơ sở tri thức, hay tri thức lĩnh vực chỉ là một phần của vấn đề độc lập lĩnh vực. Có rất nhiều phiên bản của độc lập lĩnh vực cho engine dạy học và giao diện hệ thống. Trong ESS, một engine giao diện tổng quát được cung cấp các chức năng cùng với một cơ sở tri thức đã được định nghĩa trước. Một khái niệm tương tự có thể được áp dụng cho ITSAT. Nếu engine dạy học có thể dựa vào kĩ thuật biểu diễn tri thức có liên quan nó cũng có thể được giữ độc lập lĩnh vực. Một cách tương tự điều này có thể được áp dụng cho những chiến lược dạy học thông qua tri thức lĩnh vực (Hình 2.3).





### H2.3 Những thành phần với sự phụ thuộc và sự độc lập lĩnh vực

Không may là, một công cụ phát triển ITS độc lập lĩnh vực đích thực có lẽ khó có thể đạt được bởi vì quá khó để thiết kế một công cụ có thể cung cấp cả tri thức từ nhiều lĩnh vực khác nhau và cũng phải cung cấp một tri thức sâu về bất kỳ lĩnh vực nào được đưa ra. Một sự thoả hiệp là để xây dựng một vỏ khung ITS với một lớp những lĩnh vực để một vài những hệ thống dạy học chuyên về lĩnh vực tương tự có thể được phát triển. Vì vậy, phạm vi của sự phát triển vỏ khung có thể bị giới hạn về quy mô và được tập trung nhiều hơn vào lĩnh vực liên quan.

## **2.4/ Những yêu cầu của một ITSAT :**

Việc xây dựng những công cụ soạn thảo ITS là một nhiệm vụ cực kỳ phức tạp. Để cố gắng xây dựng một công cụ mà có thể sử dụng trong bất kỳ lĩnh vực nào với nhiều chiến lược dạy học là điều không thể. Sự khác nhau của những ITS được yêu cầu là quá lớn. Để chống lại điều này, nhiều ITSAT tập trung vào một lĩnh vực đặc biệt, loại lĩnh vực hay loại dạy học. Ngay cả khi điều này được làm, có một vài phiên bản của sự phát triển ITSAT mà yêu cầu một sự xem xét chuyên biệt. Những điều này là việc tái sử dụng những thành phần hệ thống, mức độ kiểm soát hệ thống, việc dùng những chú thích bằng hình ảnh, việc xây dựng những giao diện lĩnh vực và tính tổng quát của những hệ thống đang được phát triển. Một ITSAT nên có những yêu cầu sau.

### **2.4.1/ Tái sử dụng :**

Một trong những vấn đề nền tảng trong việc phát triển một ITS là có rất ít hay không có việc tái sử dụng những thành phần giữa những sự phát triển ITS. Do đó không có sự giảm trong chi phí xây dựng ITS giữa những hệ thống khi mỗi cái chỉ xây dựng cho riêng nó trong quá trình phát triển. Nhiều thành phần của những ITS cho mượn chính chúng để tái sử dụng. Chẳng hạn, những thành phần cơ sở tri thức và những chiến lược dạy học.

Một vài thuận lợi của việc phát triển phần mềm với vấn đề tái sử dụng, gồm :

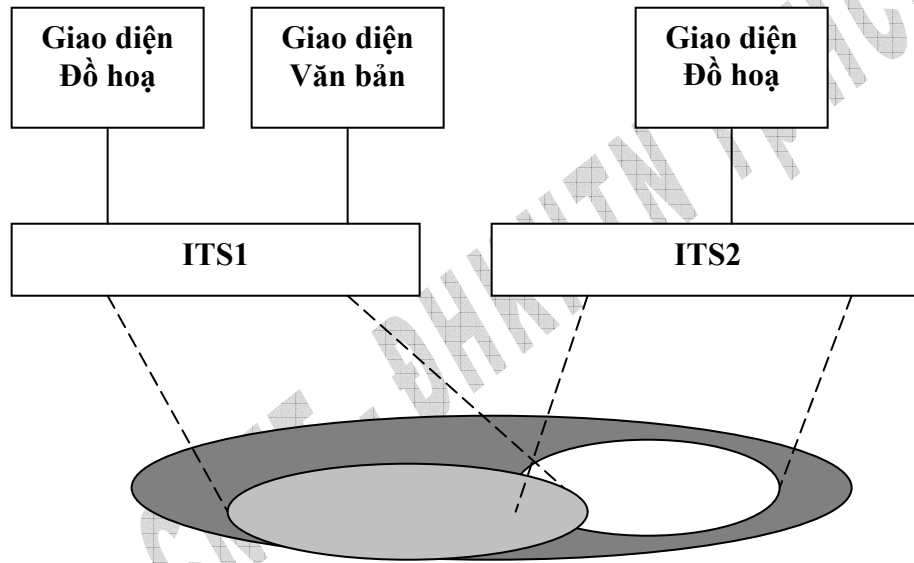
- + Tăng khả năng tin cậy của hệ thống khi sử dụng lại những thành phần đã được kiểm chứng.
- + Tạo nên việc sử dụng hiệu quả các chuyên gia bởi vì họ có thể tổng hợp tri thức của họ trong những đơn vị tái sử dụng.
- + Giảm thời gian phát triển phần mềm cũng như cả thời gian phát triển và cập nhật cũng sẽ giảm.

+ Hỗ trợ những tiêu chuẩn có tổ chức bởi vì những thành phần sử dụng lại có thể được thực thi trong một tập những tiêu chuẩn.

Việc tái sử dụng những thành phần ITS là một chức năng nền tảng của hầu hết những công cụ soạn thảo ITS. Bởi vì nhiều thành phần của những ITS là tốn nhiều thời gian và nhân lực để xây dựng theo nguyên gốc nên việc tái sử dụng càng nhiều càng tốt các thành phần là điều cần được ưu tiên. Sự sản xuất phần mềm dạy học có thể được cải thiện đáng kể bởi việc tái sử dụng những kết quả của các dự án trước. Những thành phần đặc trưng để được tái sử dụng là những cơ sở tri thức lĩnh vực, những chiến lược dạy học và sự hiển thị bằng các phương tiện đa truyền thông.

Như đã chú thích ở mục 2.3, việc tái sử dụng các cơ sở tri thức đã khuyến khích trong việc phát triển ESS. Và với cơ sở tri thức ITS, sự đạt được và sự phát triển cơ sở tri thức hệ chuyên gia là cực kỳ tốn kém, đặc biệt trong những phần sử dụng thời gian các chuyên gia lĩnh vực. Một bước quan trọng trong khả năng có thể tái sử dụng cơ sở tri thức là tiến trình phân chia cơ sở tri thức.

Một cách mà điều này có thực hiện là bởi sự phân chia tri thức lĩnh vực từ sự biểu diễn của nó. Sự phân chia này hỗ trợ việc soạn thảo bởi việc tái sử dụng những “vật liệu” đã tồn tại. Nếu tri thức lĩnh vực có thể được giữ độc lập so với kỹ thuật biểu diễn, đối với một ITS, thì một loạt những phương pháp biểu diễn khác nhau có thể được xây dựng thông qua tri thức lĩnh vực, và đối với nhiều ITS, cơ sở tri thức có thể được sử dụng dễ dàng cho những hệ thống tương lai dựa vào lĩnh vực tương tự (Hình 2.4). Tính linh hoạt này có thể thêm vào sức mạnh của một ITSAT nếu nó được nối kết với việc tái sử dụng kỹ thuật biểu diễn.



### Tri thức Lĩnh vực

#### H2.4 Chia sẻ tri thức lĩnh vực độc lập

Gần đây, với việc gia tăng việc sử dụng các “vật liệu” dựa trên phương tiện đa truyền thông, để có thể làm được và mở rộng một cách kinh tế, những vỏ khung soạn thảo cho ITS phải có thể tái sử dụng phương tiện đa truyền thông để hướng dẫn. Để làm điều này có thể cần có sự phân chia giữa các phương tiện đa truyền thông, tính chất tương tác của “vật liệu” dạy học và mạng ngữ nghĩa hướng dẫn của nó.

Nhiều sản phẩm phương tiện đa truyền thông đã có sẵn dưới những định dạng tiêu chuẩn hóa. Chẳng hạn, những đoạn phim trong QuickTime, định dạng phim MPEG và AVI, những định dạng hình ảnh GIF, JPEG, và PICT, và định dạng âm thanh WAVE, AV và MOD. Mối quan tâm hiện tại là trong việc xây dựng dựa vào tiêu chuẩn hiện tại được cung cấp trên World Wide Web (WWW) bởi HTML (HyperText Markup Language) và việc sử dụng những tính năng WWW. Hầu hết những ITS đòi hỏi những nền chuyển giao đặc biệt và rằng việc sử dụng WWW và HTML có thể cung cấp một nguồn chuyển giao độc lập nền. Một vài nhà nghiên cứu

đang nghiên cứu việc sử dụng www như là một trung gian cho sự biểu diễn ITS và nếu điều này khả thi việc tái sử dụng tài liệu HTML có thể làm giảm đáng kể việc phát triển “vật liệu” biểu diễn mà cần thiết cho những ITS mới. Và gần đây nhất là sự xuất hiện ngày càng nhiều ngôn ngữ XML (Extensible Markup Language) trong việc lưu trữ dữ liệu, xử lý ... Đặc biệt ngôn ngữ ở dạng văn bản nên có thể được sử dụng bất cứ ở đâu, không phụ thuộc nền.

Một cách đặc trưng khác mà biểu diễn bằng phương tiện đa truyền thông có thể được tái sử dụng là trong những công cụ xây dựng giao diện mà có thể được sử dụng để xây dựng giao diện người dùng. Những thư viện của những thành phần có thể được biểu diễn như là một tài nguyên bộ công cụ cho những hệ thống dạy học để tập hợp. Cung cấp những công cụ để làm dễ dàng việc xây dựng thành phần giao diện của một ITS là hiệu quả, không có những giao diện thích hợp việc học tập có thể giảm xuống, đặc biệt nếu người dùng có thể sẽ dễ dàng đánh mất sự thích thú với việc học.

Một mối quan tâm thứ ba là việc tái sử dụng chiến lược dạy học. Tri thức dạy học hay những chiến lược dạy học thường được mã hoá cứng trong những ứng dụng giáo dục. Điều này làm phức tạp việc tái sử dụng tri thức đó trong những hệ thống tiếp theo. Vấn đề này bị gây ra bởi việc thiếu những phương pháp luận được kiểm chứng cho việc xây dựng ITS. Nối kết những chiến lược dạy học và sự kiểm soát hệ thống nếu không dự tính trước làm cho khó khăn để phân chia những thành phần cho khả năng tái sử dụng. Bằng cách định danh và hệ thống hoá những chiến lược dạy học tổng quát chúng ta có thể tái sử dụng tri thức dạy học trong việc xây dựng phần mềm giáo dục.

#### **2.4.2/ Kiểm soát hệ thống :**

Một vấn đề gặp phải trong việc phát triển một công cụ soạn thảo ITS là mức độ kiểm soát mà hệ thống nên có qua sự phát triển những thành phần ITS. Một sự xem xét

quan trọng là với quyết định hướng dẫn then chốt nào dự định sẽ là những hành vi mặc định của ITSAT và bao nhiêu sự kiểm soát đối với tác giả soạn thảo.

Bởi vì một trong những mục đích của việc phát triển ITSAT là để làm nhẹ bớt cho những tác giả soạn thảo với những tác vụ mà họ không tinh thông, vấn đề tổng quát nảy sinh một lần nữa. Nếu một công cụ có tính tổng quát thích hợp, làm cách nào để nó có thể đối phó với những nhu cầu cá nhân của những tác giả soạn thảo ITS khác nhau? Cũng vậy, có vấn đề cho một ITS tổng quát phải có những tương tác thích hợp với nhiều người dùng khác nhau. Thành công của một công cụ hệ thống dạy học sẽ phụ thuộc vào khả năng ứng dụng tổng quát của nó, tính dễ dàng của việc xây dựng ứng dụng mới, cũng như khả năng để cung cấp cho người học một “nhiệm vụ” hấp dẫn và một ngữ cảnh thực tế mà họ có thể thực hiện những kỹ năng của mình.

Một hướng tiếp cận để tăng tính tiện lợi cho những nhà phát triển hướng dẫn là đặt hướng dẫn, giao diện, thiết kế và lập trình dưới sự kiểm soát của công cụ phát triển. Trong hệ thống của Hsieh và Redfield, AIDA, nhà hướng dẫn xây dựng một cơ sở dữ liệu mô tả một nhóm thiết bị hay hệ thống và phát triển và chèn những hình ảnh để minh họa thiết bị; AIDA sau đó sử dụng tri thức thiết bị trong Cơ sở dữ liệu để phát sinh lời hướng dẫn tựa dạy học với những câu hỏi thực hành và phản hồi gắn vào. Tiến trình này được làm dễ dàng bởi công cụ soạn thảo bằng cách chuyên biệt hoá thuật ngữ và những khái niệm được hiển thị bởi hệ thống đến những khái niệm từ lĩnh vực hiện tại.

Một giải pháp cho tính phức tạp của việc xây dựng những ITS là phải giới hạn phạm vi soạn thảo. Trong hướng tiếp cận này, tri thức chung trong một danh mục được mã hoá như những mô tả độc lập lĩnh vực. Những sự mô tả này sau đó được sử dụng bởi một công cụ soạn thảo trong một dạng dễ hiểu cho tác giả soạn thảo ITS. Những ví dụ của tri thức chung từ những hệ thống ITS đã tồn tại có thể được tìm thấy trong lĩnh vực lập chương trình máy tính dạy học và lĩnh vực truy vấn thắc mắc.

Một mức khác của sự kiểm soát mà một hệ thống soạn thảo có thể có qua quá trình xây dựng là theo dõi tiến trình soạn thảo. Nếu phạm vi của lĩnh vực được giới hạn, hệ thống soạn thảo có thể theo dõi và cung cấp phản hồi cho tác giả soạn thảo dưới dạng định danh những lỗi có thể và sự mâu thuẫn trong các mô tả hiện tại. Những công cụ này dò tìm và chuẩn đoán công việc của tác giả soạn thảo, theo vết những vấn đề xảy ra, chẳng hạn, sự hoàn thiện của phần mềm dạy học và phần ngữ pháp chứa đựng những mẫu có tính đàm thoại không liên kết. Hệ thống này chứa một thành phần “cố vấn” mà được sử dụng để giúp đỡ người thiết kế sử dụng một tập những công cụ phát triển hệ thống. Những quy tắc cho việc sử dụng hiệu quả những công cụ được sử dụng để trợ giúp tác giả soạn thảo. Chẳng hạn, nếu mức kỹ năng của ITS không thích hợp với những nhu cầu huấn luyện được định danh trong pha phân tích thì một quy tắc sẽ được “bắt” để hiển thị một thông điệp cảnh báo thích hợp.

#### **2.4.3/ Những chú thích bằng hình ảnh cho sự đạt được tri thức :**

Thông tin bằng hình ảnh, đặc biệt là những thông tin có bản chất rắc rối và phức tạp, trong nhiều năm là chủ đề của công việc có thể xem xét bởi nhiều người. Việc sử dụng những chú thích bằng hình ảnh có thể đặc biệt có ích cho việc đạt được tri thức. Khi nhiều người sử dụng tiềm năng những ITSAT có rất ít hay không có sự hiểu biết về những kỹ thuật biểu diễn tri thức, việc sử dụng chú thích bằng đồ họa có thể làm dễ dàng những yêu cầu học tập của họ cho pha đạt được tri thức của sự phát triển một ITS.

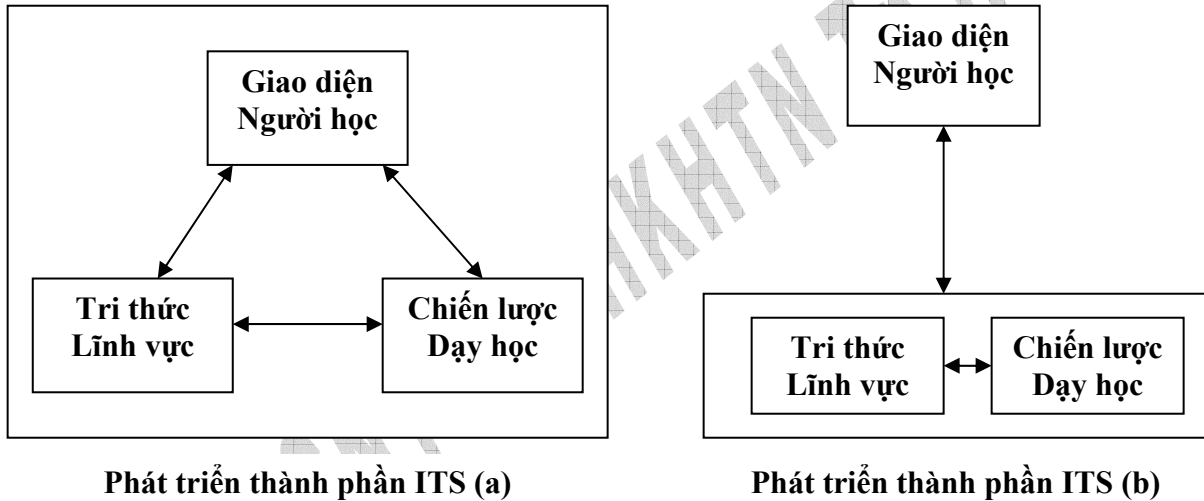
Từ quan điểm của một ITSAT, việc cung cấp một chú thích bằng hình ảnh phụ thuộc vào 2 thứ. Đầu tiên là, kỹ thuật biểu diễn tri thức là phải được sử dụng; và thứ hai là, loại giao diện được sử dụng bởi hệ thống ITSAT. Việc lựa chọn một loại giao diện phức tạp bởi các phiên bản của nền chuyển giao hệ thống và các tài nguyên hệ thống.

#### **2.4.4/ Phát triển những giao diện lĩnh vực :**

Bởi việc giảm giá thành trong phần cứng về đồ hoạ (Màn hình, CDROM, đĩa laser, và các thiết bị nhập trực tiếp) và tính sẵn sàng của những công cụ này, không có một hạn chế về những giao diện người dùng mà có thể được tạo ra. Vì vậy, chỉ có một giới hạn thực sự về việc xây dựng giao diện là thời gian để xây dựng giao diện và mức độ chi tiết mà được yêu cầu cho dạy học được trình bày. Sự kết hợp hài hoà của các phương tiện đa truyền thông đối với ITS cung cấp tiềm năng để tạo ra những hệ thống thông minh mà có thể hướng dẫn và minh hoạ bằng cách sử dụng âm thanh, hoạt hình và đoạn phim.

Không may là việc xây dựng giao diện người dùng thường nằm ngoài sự hiểu biết của nhiều tác giả soạn thảo ITS sau này. Cung cấp một bộ công cụ ở mức cao cho thiết kế giao diện và phát triển có thể làm giảm nhu cầu cho tri thức xây dựng giao diện lớn. Điều này hữu ích khi một tiến trình chính trong thiết kế và xây dựng một ITS đang xây dựng một giao diện cho người học sử dụng. Hai hướng tiếp cận cho việc phát triển những giao diện có thể thấy ở Hình 3.5. Trong Hình 3.5(a), nội dung lĩnh vực được liên kết trực tiếp đến giao diện người dùng và trong Hình 3.5(b), nội dung lĩnh vực được giữ riêng với sự phát triển giao diện.





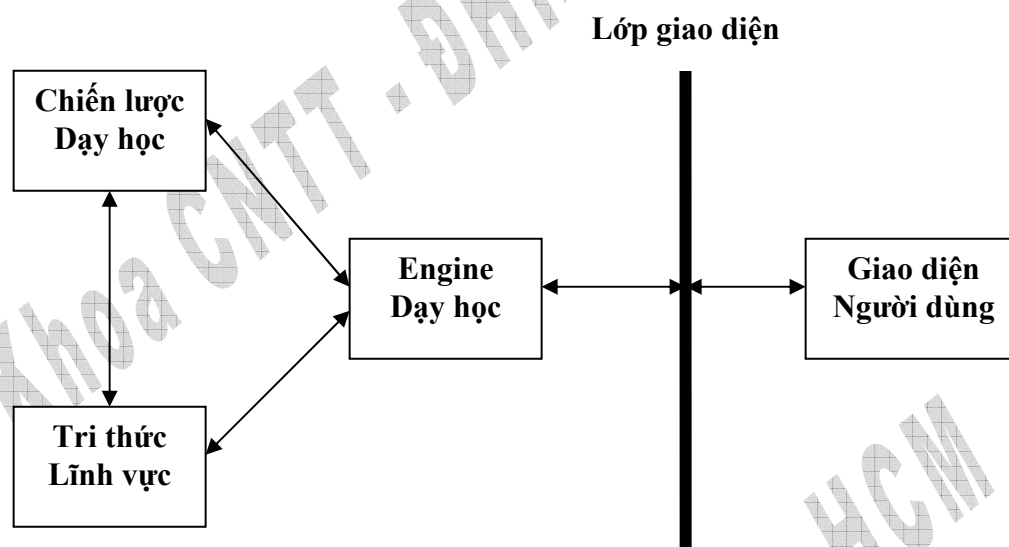
### H2.5 Hai hướng tiếp cận đến phát triển giao diện

Trong nhiều trường hợp việc thiết kế giao diện người học được kết hợp với việc thiết kế môn học và bài học. Điều này có thể dẫn đến trong những công cụ phát triển không lỗi mà thỉnh thoảng khó sử dụng và có thể không linh hoạt trong ứng dụng của chúng.

Một trong những thuận lợi chính của việc giữ tách rời việc phát triển giao diện là để không có giới hạn nào về loại giao diện có thể được sử dụng. Nếu mục đích của ITSAT là càng tổng quát càng tốt, thì cung cấp khả năng linh hoạt trong việc lựa chọn giao diện là rất quan trọng. Một cách lý tưởng, sự cung cấp để bất kỳ công cụ phát triển giao diện nào có thể được sử dụng để xây dựng giao diện người dùng là điều được mong muốn.

Tuy nhiên, hướng phân chia giao diện từ lĩnh vực không có thuận lợi riêng của nó. Một công cụ giao diện độc lập không có ngữ cảnh để dựa vào sự phát triển của nó trong đó. Mỗi một giao diện phải được hoàn chỉnh ngay từ đầu. Nếu một sự phát triển

giao diện được liên kết với một lĩnh vực thì sau đó những thành phần biểu diễn lĩnh vực có thể được “thu thập” vào trong một thư viện tài nguyên để hỗ trợ việc phát triển giao diện. Điều này khuyến khích việc tái sử dụng những thành phần giao diện cho ITS trong những lĩnh vực tương tự. Trong khi việc tái sử dụng này là mong muốn, chi phí linh hoạt trong việc thiết kế giao diện.



**H2.6 Sự phân chia giao diện ITS**

Trong công việc hiện tại, hướng thứ hai đã được thực hiện. Lợi ích tiềm năng trong khả năng linh hoạt và tự do trong việc lựa chọn giao diện và những thuận lợi của việc giữ những thành phần của một ITS độc lập dường như được thanh minh. Để cho phép một giao diện độc lập lĩnh vực để được thêm một ITS ở đây là cần một vài tầng tiêu chuẩn hóa giữa giao diện và hệ thống ITS (Hình 2.6).

**2.4.5/ Tính tổng quát và riêng biệt :**

+ Khả năng ứng dụng tổng quát của một ITSAT có thể trở thành mối quan tâm lớn trong việc phát triển của nó. Đây là điều được mong muốn nếu một ITSAT hoàn

toàn có thể được phát triển. Không may là, có nhiều mức tổng quát khác nhau khi xây dựng ITS mà cần xem xét. Đầu tiên, có tính tổng quát của ITS hoàn chỉnh. Độc lập lĩnh vực được mong muốn cho một ITS nhưng thật khó để đạt được mà không giới hạn sức mạnh dạy học của hệ thống. Thứ hai là, kiểu của những chiến lược dạy học mà có thể được hỗ trợ và định nghĩa sẽ mất đi chất lượng nếu một hướng tiếp cận tổng quát được thực hiện. Thứ ba là, một công cụ tổng quát phải cung cấp một giao diện tổng quát để áp dụng cho một loạt nhiều tác vụ. Bởi tính tổng quát của nó, giao diện phải độc lập lĩnh vực và vì vậy được giới hạn. Những công cụ mà nhằm vào tính tổng quát có tiềm năng để hỗ trợ việc thiết kế một loạt nhiều ứng dụng nhưng thường áp đặt những sự giới hạn vào kiểu tương tác trong ứng dụng nhằm tới, cung cấp ít sự hỗ trợ thiết kế và dựa vào những mô hình hướng dẫn tổng quát.

Mặc dù một mức độ cao của tính tổng quát có thể dường như được mong muốn, một sự thực thi tốt của những công cụ soạn thảo như vậy sẽ cực kỳ khó khăn để đạt được. Cố gắng cung cấp một công cụ với mức độ của tính tổng quát dẫn đến cho ITS hoặc là yếu về mặt dạy học hoặc là đòi hỏi những nỗ lực phát triển cao.

+ Có lẽ không có lời khuyên nào để xây dựng những ITSAT có mục đích tổng quát, nhưng tập trung vào những công cụ với những chức năng tổng quát cho một lĩnh vực riêng biệt hay loại tác vụ là một hướng tiếp cận tốt. Hướng tiếp cận này có thể cung cấp những công cụ với mục đích đặc biệt hơn là những mô hình chung và được xây dựng dựa vào một mô hình rõ ràng của việc xây dựng. Khi môi trường được tập trung vào một lĩnh vực, những công cụ này nhiều khả năng hơn để có thể cung cấp một môi trường phát triển tương tác giàu tiềm năng, và sẽ tốt hơn để có thể cung cấp sự hướng dẫn thiết kế. Những bất lợi chính cho hướng tiếp cận này là có một dãy các thiết kế hẹp mà có thể được phát triển và có sự phụ thuộc vào mô hình mà được “xúc tiến” bởi công cụ.

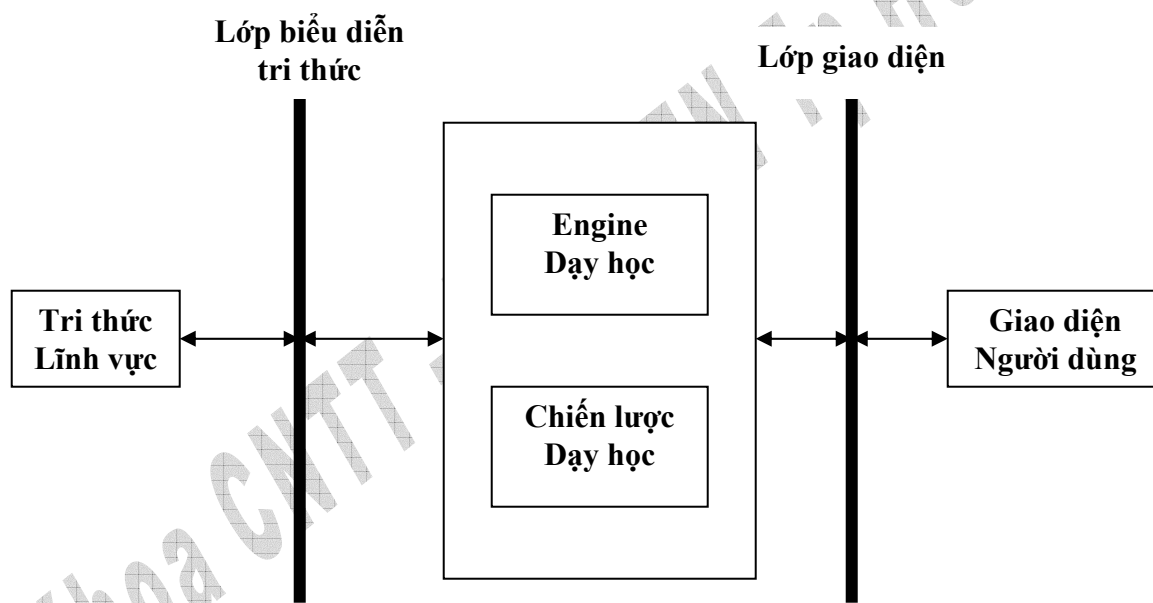
Cũng vậy bằng cách chuyên biệt hoá những công cụ, mỗi cái hỗ trợ việc thiết kế của một loại phần mềm giáo dục riêng biệt. Điều này có thuận lợi là tri thức thiết kế dễ dàng mã hoá, bởi vì phạm vi hẹp hơn và những người dùng và giao diện tiềm năng được đảm bảo bây giờ có lẽ đặc biệt hoá hơn cho những tác vụ liên quan. Không may là, những công cụ như vậy đánh mất tính tiện ích của chúng.

Một cách mà sự chuyên biệt hoá có thể đạt được là bằng cách giới hạn môi trường học tập, chẳng hạn những kiểu của các chiến lược dạy học được tận dụng, hay những kĩ thuật biểu diễn có thể được định nghĩa dựa vào một vài tập mô hình nhận thức.

+ Hướng tiếp cận thứ ba là sự kết hợp của hướng tiếp cận chuyên biệt và hướng tiếp cận tổng quát. Để làm điều này, một định danh của những thành phần của một ITS mà sẽ hưởng lợi hầu như từ việc chuyên biệt hay tổng quát được cần thiết.

Một lần nữa những phiên bản của thông tin nào thì độc lập lĩnh vực và cái gì thì phụ thuộc phải được đưa vào bảng kê khai. Như đã nói ở những phần trước, hai thành phần phụ thuộc lĩnh vực chính của một ITS là **tri thức lĩnh vực** và **giao diện lĩnh vực**. Điều này để cho thành phần kiểm soát/dạy học và những chiến lược dạy học mà phải được sử dụng. Nếu những điều này có thể được giữ độc lập lĩnh vực thì sau đó một nửa mô hình ITS chuẩn có thể hoàn thành yêu cầu độc lập lĩnh vực mà thúc đẩy việc tái sử dụng của những thành phần và tính tổng quát của hệ thống cuối cùng (Hình 2.7).

Hướng tiếp cận này được khuyến khích nếu những công cụ phân chia được sử dụng cho việc đạt được tri thức và sự phát triển giao diện người dùng. Vì vậy làm đơn giản sự phát triển của những ITS tương lai dựa vào môi trường giống nhau bởi việc tái sử dụng những thành phần độc lập lĩnh vực.



**Những thành phần  
độc lập lĩnh vực**

**H2.7 Phân chia những thành phần ITS**

Để giới hạn phạm vi của công việc hiện tại với những gì đã được thảo luận ở hướng tiếp cận thứ ba, 3 thành phần của ITSAT được chuyên biệt hoá: kĩ thuật biểu diễn tri thức, môi trường dạy học và loại vấn đề mà có thể được dạy, cụ thể là việc dạy những kĩ năng thủ tục.

**2.5/ Dạy học những kĩ năng thủ tục :**

Dạy người sử dụng để điều hành các thiết bị hay để học những thủ tục phức tạp có thể tốn nhiều thời gian và chi phí. Đây là trường hợp đặc biệt khi thiết bị thực hành thì tốn kém và dễ hỏng. Cũng vậy, nếu thiết bị phải hoạt động trong môi trường nguy hiểm, việc huấn luyện dựa trên công việc có lẽ là không thể. Một giải pháp cho những vấn đề này là sử dụng những môi trường giả lập dựa vào máy tính.

Có một vài thuận lợi khi sử dụng những giả lập trên máy tính như nền tảng của một môi trường học tập. Đầu tiên là, “ứng xử dụng cụ” có thể được định nghĩa một cách chính xác. Thế giới thực thường cung cấp nhiều ảnh hưởng không thể dự đoán và những yếu tố ngẫu nhiên mà có thể có hại nếu việc học trong một tình huống tập trung được yêu cầu. Thứ hai là, nhiều môi trường hoặc là quá nguy hiểm hoặc là có thể dẫn tới những kết quả tai hại không lường trước. Chẳng hạn, khi huấn luyện một kỹ sư trong hoạt động phản ứng hạt nhân, huấn luyện thực hành có lẽ không thể chấp nhận. Cũng vậy, nếu “dụng cụ” hay “vật liệu” được sử dụng là rất đắt, huấn luyện dựa trên công việc có thể là rất tốn kém.

Cũng như với các huấn luyện dựa trên máy tính khác, việc sử dụng những giả lập trên máy tính có thuận lợi trong việc cung cấp việc dạy cho nhiều người dùng vào cùng một thời điểm và một khi những gói được phát triển chúng có thể được tái sử dụng khi cần thiết. Vì vậy huấn luyện đòi hỏi việc dạy “một kèm một” có thể là sẵn sàng hơn. Không may là, việc xây dựng một môi trường giả lập thích hợp không phải là một công việc dễ dàng.

Huấn luyện trong các ngành công nghiệp dựa trên máy tính có hai hướng tiếp cận. Đầu tiên là phát triển sự giả lập bản sao đúng như thật sử dụng sự huấn luyện dựa trên thí nghiệm thực tế với người quản trị là người thật. Loại giả lập này là rất đắt, cả trong quá trình phát triển và sự quản trị, nhưng có thuận lợi trong việc cung cấp những môi trường huấn luyện hiện thực và cho phép sự thử nghiệm bởi nó khó thực hiện do những lý do an toàn. Hướng tiếp cận khác là hiển thị một tác vụ trên máy tính và kiểm tra người được huấn luyện tính thực thi đúng của tác vụ. Mặc dù môi trường này hạn chế hơn hướng tiếp cận đầu tiên, nhưng nó có thể rất hiệu quả cho các mục đích huấn luyện dựa trên thủ tục đã được định nghĩa tốt.

Phương pháp thứ hai của việc dạy này được tìm thấy tính hiệu quả một cách đặc biệt khi một tiến trình được giả lập và người học hoặc là tương tác với hệ thống mô

hình hoá hoặc là phải cố gắng để sửa sai các vấn đề khi một tiến trình không hoạt động chính xác. Khi dạy những kỹ năng thủ tục trong một ngữ cảnh giải quyết vấn đề, người học xuất hiện để học những thông tin một cách có hiệu quả hơn nếu họ được hiện diện với những thông tin đó xuyên suốt quá trình giải quyết vấn đề hơn là được dạy những nguyên tắc chung chung.

Dạy học những kỹ năng thủ tục là một khía cạnh quan trọng của việc nghiên cứu về những hệ thống dạy học thông minh. Những phương pháp như vậy đã được sử dụng để tạo ra những loại môi trường học tập khám phá trong một loạt những lĩnh vực khác nhau.

Công việc hiện tại đang tập trung vào sử dụng những kỹ thuật để dạy những kỹ năng thủ tục trong một môi trường giả lập dựa vào máy tính. Điểm nhấn mạnh chính là về những tác vụ mà có các bước tuần tự với một vài thứ tự có thể, đặc biệt trong lĩnh vực bảo trì và thao tác dụng cụ thiết bị.

Tuy nhiên, khi cung cấp một môi trường giả lập cho một lĩnh vực thì cần có một vài giới hạn về sự giả lập để cho phép phạm vi của mô hình để định nghĩa. Không có bất kỳ giới hạn về những mô hình giả lập, chúng sẽ trở nên lớn, và sự sản xuất và điều khiển của chúng sẽ không thể quản lý. Giới hạn chính đối với công việc hiện tại là việc sử dụng một giả định thể giới tĩnh để giữ cho nó ở kích thước và chức năng có thể quản lý được.

## **2.6/ Giới thiệu một vài công cụ soạn thảo :**

Phần trên ta đã trình bày khái niệm cũng như một vài yêu cầu mà một ITSAT cần có. Trong phần này, ta sẽ trình bày tiếp một vài công cụ soạn thảo đã hoàn chỉnh. Đầu tiên, đó là sự phân loại công cụ soạn thảo. Sự phân loại này chỉ nhằm phân biệt những ITSAT đã phát triển, không nhằm đưa ra một sự phân loại tổng quát cho tất cả các công cụ soạn thảo. Vì vậy đây chỉ là một căn cứ để các nhà phát triển có thể dựa

vào để phát triển những hệ thống của mình. Cuối phần này ta sẽ mô tả nội dung của một ITS cần soạn thảo, cũng như các phương pháp soạn thảo.

**2.6.1/ Sự phân loại theo tác vụ và dạy học :**

Bất kỳ sự thảo luận nào về công cụ sẽ là quá trừu tượng mà không có một vài ngữ cảnh miêu tả những hệ thống dạy học mà chúng được sử dụng để xây dựng. ITSAT được sử dụng để xây dựng những hệ thống dạy học trong nhiều lĩnh vực, bao gồm dịch vụ khách hàng, toán học, bảo trì thiết bị... Những hệ thống dạy học này hướng đến mục đích là nhiều đối tượng người học. Tuy nhiên, sự khác nhau quan trọng giữa các ITSAT không liên quan đến lĩnh vực chuyên biệt hay số lượng người học, nhưng đến những khả năng độc lập lĩnh vực mà những ITS đã soạn thảo có. Trong phần này, ta trình bày một sự phân loại những công cụ soạn thảo dựa trên khả năng của chúng.

Bảng 1.1 liệt kê 7 danh mục ITSAT, được nhóm theo loại ITS mà chúng tạo ra. Bảng 2.2 tóm tắt những phần bên dưới, mà miêu tả sức mạnh và giới hạn của mỗi loại công cụ soạn thảo, và sự khác nhau giữa những công cụ soạn thảo trong các nhóm. Bảng 2.2 mô tả sức mạnh và sự giới hạn của mỗi danh mục, cùng với sự tóm tắt về những hệ thống trong danh mục khác nhau như thế nào.

**Bảng 1.1 Danh mục phân loại những ITSAT**

	<b>Danh mục</b>	<b>Hệ thống ví dụ</b>
1	Sự sắp thứ tự và lên kế hoạch môn học	DOCENT, IDE, ISD Expert, Expert CML
2	Những chiến lược dạy học	Eon, GTE, REDEEM
3	Giả lập thiết bị và huấn luyện thiết	DIAG, RIDES, SIMQUEST, XAIDA



	bị	
4	Hệ chuyên gia lĩnh vực	Demonstr8, D3 Trainer, Training Express
5	Loại đa tri thức	CREAM-Tools, DNA, ID-Expert, IRIS, XAIDA
6	Mục đích đặc biệt	IDLE-Tool/Imap, LAT
7	Chương trình được dạy học dựa trên media thích ứng / thông minh	CALAT, GETMAS, InterBook, MetaLinks

**Bảng 2.2 : Sức mạnh và giới hạn của những ITSAT theo danh mục**

Danh mục	Sức mạnh	Giới hạn	Sự khác nhau
Sự sắp thứ tự và lên kế hoạch môn học	Những quy tắc, sự ràng buộc hay chiến lược cho việc sắp thứ tự các khoá học, modul, sự trình bày	Tính đúng đắn thấp, sự biểu diễn kỹ năng “cạn”	Hoặc là những quy tắc sắp thứ tự được sửa hoặc là được soạn; “vật liệu” làm nền của quá trình soạn thảo
Những chiến lược dạy học	Những chiến lược dạy học ở mức rất nhỏ; tập các hướng dẫn cơ bản rất tinh	(nt)	Phương pháp biểu diễn chiến lược; nguồn của chuyên môn hướng dẫn

	vi; đa chiến lược dạy học		
Giả lập thiết bị và huấn luyện thiết bị	Sự soạn thảo và dạy học phù hợp với sự định danh, hoạt động và truy vấn thắc mắc thành phần thiết bị	Những chiến lược hướng dẫn giới hạn; mô hình người học giới hạn; hầu hết là những kỹ năng thủ tục	Tính đúng đắn của sự giả lập; dễ soạn thảo
Hệ chuyên gia lĩnh vực	Mô hình chuyên môn lĩnh vực có thể chạy; mô hình và sự chuẩn đoán người học tốt; bao gồm những quy tắc cho người mới học và những quy tắc bắt lỗi	Xây dựng hệ chuyên gia khó; giới hạn đối với chuyên môn giải quyết vấn đề và thủ tục; giới hạn những chiến lược hướng dẫn	Những mô hình nhận thức hay thực thi chuyên môn
Loại đa tri thức	Sự biểu diễn rõ ràng và những phương pháp hướng dẫn được định nghĩa trước cho các sự kiện, khái niệm và thủ tục	Giới hạn đến những sự kiện, khái niệm, thủ tục đơn giản có liên quan; những chiến lược dạy học đã được định nghĩa trước	Bao gồm sự sắp thứ tự môn học thông minh; loại tri thức /tác vụ được hỗ trợ

Mục đích đặc biệt	Những hệ thống được dựa trên mẫu cung cấp hướng dẫn soạn thảo mạnh, thiết kế từng phần hay những nguyên tắc có liên quan đến dạy học có thể được tuân theo.	Mỗi công cụ giới hạn cho một loại hệ thống dạy học riêng biệt; tính cứng của sự biểu diễn và dạy học .	Mức độ của tính cứng
Chương trình dạy học dựa trên media thích ứng / thông minh	WWW có khả năng truy cập và giao diện người dùng đồng nhất; sự lựa chọn thích ứng và lời chú thích của những siêu liên kết	Giới hạn ở tính tương tác; Giới hạn độ rộng của mô hình người học	Sự tập trung ở mức độ vĩ mô và vi mô; mức độ của sự tương tác

Lúc đầu những ITSAT rơi vào 2 loại chính : những cái nghiêng về sự giả lập thiết bị và bao gồm một môi trường học tập các hướng dẫn; và những cái dựa trên các phần mềm truyền thống. Ngay cả mặc dù những hệ thống gần đây nối kết những khía cạnh của cả hai, nhưng đa số những công cụ soạn thảo rơi tương tự vào 2 danh mục : Hướng về giáo dục và Hướng về thực hành. Những hệ thống hướng giáo dục (danh mục 1,2,5,7 trong Bảng 1.1) tập trung vào cách sắp thứ tự và dạy những nội dung được đúng hộp có liên quan. Những hệ thống thực hành (danh mục 3,4, và 6 trong Bảng 1.1)

tập trung vào việc cung cấp một môi trường học tập “giàu chức năng”, ở đó người học có thể học tập những kỹ năng bằng cách rèn luyện và nhận những phản hồi.

Bên dưới là mô tả những danh mục được liệt kê trong Bảng 1.1 và 1.2

### **2.6.1.1/ Sự sắp thứ tự và lên kế hoạch môn học :**

Những hệ thống soạn thảo trong danh mục sắp thứ tự khoá học và môn học tổ chức những đơn vị hướng dẫn (IUs – Instruction Units) vào trong cấu trúc cây các khoá học, các modul, các bài học, các sự biểu diễn ... mà liên quan bởi các điều kiện tiên quyết, và những mối liên hệ khác. Những đơn vị hướng dẫn bình thường có những mục đích hướng dẫn. Một vài hệ thống bao gồm những đơn vị hướng dẫn mà giảng giải các khái niệm khó hiểu hay “vật liệu” dành cho người học yếu kém. Nội dung được lưu trữ trong văn bản và đồ hoạ được đóng gói. Những hệ thống này được xem như những công cụ để giúp những nhà thiết kế hướng dẫn và giáo viên thiết kế những khoá học và quản lý việc học dựa trên máy tính.

Sự sắp xếp thông tin của những đơn vị hướng dẫn (hay nội dung, hay những chủ đề) là lõi của những hệ thống này. Đối với người học, những hệ thống dạy học được xây dựng với những công cụ này có lẽ dường như giống hệt đối với sự hướng dẫn dựa trên máy tính truyền thống. Những màn hình của văn bản và hình ảnh đã đóng gói được hiển thị, và những sự tương tác có khả năng giới hạn ở dạng trắc nghiệm, điền chỗ trống,... Dĩ nhiên, sự khác nhau là sự sắp xếp nội dung được xác định một cách động dựa vào năng lực của người học, mục đích bài học, và những mối quan hệ giữa những module khoá học. Bởi vì tri thức lĩnh vực không được biểu diễn ở chiều sâu nên lĩnh vực nào cũng có thể được dạy. Nhưng chiều của sự chuẩn đoán và phản hồi trong những hệ thống dạy học được xây dựng với những công cụ soạn thảo này được giới hạn bởi độ “nông” của sự biểu diễn tri thức lĩnh vực. Điều này làm cho chúng thích hợp hơn cho việc xây dựng những hệ thống dạy học về khái niệm, khai báo và từng đoạn của tri thức, và ít mạnh mẽ xây dựng những hệ thống dạy học mà dạy những kỹ

năng giải quyết vấn đề và thủ tục. Nói một cách tổng quát, những hệ thống soạn thảo trong danh mục sắp thứ tự môn học là cơ bản nhất, hay có chức năng một cách tối thiểu (mặc dù mỗi hệ thống trong danh mục có những tính năng hay khả năng quan trọng nhất định). Một vài danh mục khác được mô tả bên dưới chứa đựng những khả năng tối thiểu (chẳng hạn sự sắp thứ tự môn học hay lên kế hoạch) và thêm vào tính năng bổ sung.

### **2.6.1.2/ Những chiến lược dạy học :**

Những hệ thống trong danh mục này làm tốt về việc biểu diễn những chiến lược dạy học khác nhau. Chúng có khả năng tương tự với những hệ thống sắp thứ tự môn học được mô tả ở trên, trong đó nội dung được lưu trữ trong văn bản và đồ họa được “đóng hộp” và sự biểu diễn tri thức lĩnh vực thì “nông cạn”. Tuy nhiên, những hệ thống này cũng mã hoá những chiến lược tốt bởi giáo viên và chuyên gia hướng dẫn. Những hệ thống trong danh mục 1 có khả năng tập trung vào mức hướng dẫn “vĩ mô”, chẳng hạn sự sắp thứ tự của những chủ đề hay module, trong khi những hệ thống trong danh mục này tập trung vào mức hướng dẫn “vi mô”. Những quyết định hướng dẫn ở mức “vi mô” bao gồm khi nào và như thế nào để đưa những lời giải thích, tóm tắt, ví dụ và sự tương tự; loại gợi ý và phản hồi nào được cho; và loại câu hỏi và bài tập nào để đưa cho người học. Những hệ thống trong danh mục 1 có tập tính vi các chiến lược dạy học cơ bản, được so sánh với những hệ thống trong danh mục khác. Cũng vậy tính chất đối với những hệ thống trong danh mục này là khả năng biểu diễn đa chiến lược dạy học và “siêu chiến lược” mà lựa chọn chiến lược dạy học thích hợp cho tình huống đưa ra.

Cũng như trong những hệ thống trong danh mục 1, người học sử dụng những hệ thống dạy học được xây dựng với những công cụ soạn thảo này sẽ nhìn thấy những màn hình của sự tương tác bị giới hạn — chúng sẽ học bằng cách học và suy nghĩ hơn là học bằng cách làm. Tuy nhiên, tính sẵn sàng và sự xen vào thông minh của những lời giải thích, gợi ý ở các mức khác nhau, và sự tương tự có thể làm cho hệ thống dạy

học đường như sẵn sàng hồi đáp, được so sánh với sự dạy được xây dựng với hệ thống trong danh mục 1.

### **2.6.1.3/ Giả lập thiết bị và huấn luyện thiết bị :**

Đối với những hệ thống dạy học được xây dựng bởi những công cụ soạn thảo trong danh mục này, người học được trình bày một mẫu thiết bị và được yêu cầu để định danh những thành phần của nó, thực hiện những bước vận hành, thực hiện những bước bảo trì, hay chuẩn đoán hành vi lỗi thiết bị và sửa hay thay thế những phần liên quan. Những loại kỹ năng này tương đối rộng và tổng quát, vì vậy những công cụ soạn thảo được chuyên biệt hoá trong lĩnh vực này nên có thể sử dụng rộng rãi. Tri thức chuyên môn cho sự định vị thành phần và những kịch bản vận hành thì dễ dàng để mô hình hoá. Việc theo dõi thực hiện và phản hồi hướng dẫn thì cũng dễ. Vì vậy những công cụ soạn thảo có thể được xây dựng phù hợp với những nhu cầu của tác giả soạn thảo (và người học). Nhiệm vụ soạn thảo khó khăn nhất với những hệ thống này là xây dựng sự giả lập thiết bị. Nhưng một khi sự giả lập được soạn thảo, nhiều đặc điểm kỹ thuật hướng dẫn sẽ tự có. Sự định vị thành phần hay hành vi thiết bị có thể được tổng quát hoá một cách tự động. Tuy nhiên, những thủ tục vận hành thiết bị phải được soạn thảo.

Trái ngược với những công cụ soạn thảo trong 2 danh mục trước, người học sử dụng những hệ thống dạy học được xây dựng với những công cụ soạn thảo trong danh mục này sẽ được “học bằng cách làm”. Sự hướng dẫn có tính chất giới thiệu hay thuộc khái niệm sẽ vắng mặt hay bị giới hạn, và chúng thừa nhận là người học có sự thân thiện cơ bản với những khái niệm và thủ tục quan trọng trong lĩnh vực trước khi sử dụng hệ thống dạy học. Những hệ thống dạy học này là những môi trường học tập để rèn luyện những kỹ năng. Những phản hồi riêng biệt được cho đối với mỗi bước kỹ năng, và độ khó của tác vụ được tăng lên cùng với tiến trình của người học.

Nhân tố khác biệt chính giữa những hệ thống trong danh mục này là chiều sâu và sự chính xác của bộ giả lập thiết bị. Những công cụ soạn thảo “lên xuống” từ những cái hỗ trợ những nội quan hệ dựa trên sự diễn đạt tĩnh giữa những thành phần thiết bị đến những cái hỗ trợ khả năng chạy nhưng những mô hình giả lập nông cạn; đến những cái hỗ trợ độ sâu hơn, những mô hình nhận thức hay nguyên nhân của cách mà thiết bị làm việc. Những công cụ cũng thay đổi rộng trong các loại thiết bị và những tiến trình vật lý mà chúng có thể mô hình hoá.

#### **2.6.1.4/ Hệ chuyên gia lĩnh vực :**

Một lớp quan trọng của những hệ thống dạy học thông minh là những công cụ mà bao gồm những mô hình nhận thức dựa quy tắc của chuyên môn lĩnh vực. Những hệ thống dạy học như vậy, thường gọi là những hệ thống dạy học theo vết, quan sát hành vi người học và xây dựng một mô hình nhận thức tốt của tri thức người học mà có thể được so sánh với những mô hình chuyên gia. Những công cụ soạn thảo được làm nguyên mẫu cho những hệ thống dạy học như vậy. Trong danh mục này, những công cụ soạn thảo sử dụng hệ chuyên gia truyền thống ( được xây dựng để giải quyết vấn đề, không phải để dạy) và sản xuất ra sự hướng dẫn “thêm giá trị” cho chuyên môn được mã hoá. Những hệ thống này tương tự với những hệ thống truy vết theo mẫu, ngoại trừ hệ chuyên gia dựa trên khả năng thực hiện, hơn là những tiến trình nhận thức. Một vài hệ thống bao gồm những quy tắc ở mức mới học hay những quy tắc bắt lỗi mà bắt những lỗi chung, cho phép đưa ra những phản hồi đặc biệt cho những lỗi này trong quá trình dạy học.

Người học sử dụng những hệ thống này thường giải quyết những vấn đề và những vấn đề phụ liên kết trong cùng mục đích và nhận phản hồi khi những phân kỳ hành vi của họ từ mô hình chuyên gia. Không giống hầu hết những hệ thống được miêu tả ở trên, những hệ thống này có một mô hình về chuyên môn tương đối “sâu”, và vì vậy người học, khi bị cản trở, có thể yêu cầu hệ thống dạy học thực hiện bước kế tiếp,

hay để hoàn thành giải pháp cho toàn bộ vấn đề. Soạn thảo một hệ chuyên gia là một nhiệm vụ cần nhiều thời gian và khó khăn, và chỉ những tác vụ nhất định có thể được mô hình theo kiểu này.

#### **2.6.1.5/ Loại đa tri thức :**

Những lý thuyết thiết kế hướng dẫn phân loại tri thức và tác vụ vào trong những danh mục riêng biệt, và quy định những phương pháp hướng dẫn cho mỗi danh mục. Chúng có khả năng được giới hạn vào loại tri thức mà có thể dễ dàng được định nghĩa, chẳng hạn những sự kiện, những khái niệm và những thủ tục. Mặc dù những loại tri thức và những phương pháp hướng dẫn thay đổi cho những lý thuyết khác nhau, chúng quy định một cách đặc thù sự hướng dẫn tương tự những cái sau. Những sự kiện được dạy với những bài thực hành lặp lại và những thiết bị giúp trí nhớ; những khái niệm được dạy sử dụng sự tương tự và những ví dụ khẳng định và phủ định tiến triển từ nguyên mẫu; những thủ tục được dạy từng bước tại một thời điểm. Sự hướng dẫn cho những loại tri thức bao gồm cả sự biểu diễn có tính mô tả của tri thức, và những bài tập có tính dò hỏi mà cho phép sự thực hành và phản hồi. Những chiến lược hướng dẫn sẽ dễ cho việc làm thế nào để sắp thứ tự nội dung và những bài tập, và làm thế nào để cung cấp phản hồi, được định nghĩa một cách riêng lẻ cho từng loại tri thức. Bản chất của tri thức được định nghĩa trước và những chiến lược hướng dẫn là cả về sức mạnh và độ yếu của những hệ thống này. Tri thức lĩnh vực cho từng loại tri thức có thể được biểu diễn dễ dàng cho những tác giả soạn thảo, người điền chỗ trống theo mẫu cho những ví dụ, những bước, những định nghĩa, ...(phụ thuộc vào loại tri thức). Những công cụ này hỗ trợ sự phân tích của những kỹ năng phức tạp trong những thành phần tri thức cơ sở, và liên kết giữa những thành phần tri thức có thể được soạn thảo và sử dụng trong sự hướng dẫn (chẳng hạn những khái niệm và sự kiện mà hỗ trợ một thủ tục và một khái niệm mà giúp giải thích một khái niệm khác). Bởi vì những chiến lược hướng dẫn được sửa và dựa vào những loại tri thức nên chúng không phải được soạn



thảo. Dĩ nhiên, không phải tất cả sự hướng dẫn đều phù hợp với nền tảng này, nhưng nó có tính ứng dụng thật sự rộng.

Những công cụ soạn thảo trong danh mục này thay đổi theo nhiều khía cạnh, nhưng tất cả chúng sử dụng sơ đồ phân loại kỹ năng/tri thức hiển thị và hướng dẫn một cách khác nhau dựa vào loại tri thức. Cũng vậy, chúng có thể trích dẫn những tài liệu thiết kế hướng dẫn cổ điển như phần của nền tảng cho hướng tiếp cận giáo dục của chúng. Đối với người học, những hệ thống dạy học được xây dựng với những hệ thống này tương tự vì bản chất của những cái trong danh mục 2. Sự khác nhau chính là đối với người soạn thảo, nhiệm vụ của họ là phụ thuộc hơn, và vì vậy cả dễ dàng hơn và ít linh động hơn.

#### **2.6.1.6/ Mục đích đặc biệt :**

Trong danh mục này thì những công cụ soạn thảo được chuyên biệt hoá trong những tác vụ hay lĩnh vực đặc biệt. Những hệ thống trong danh mục 3 cũng cho những loại tác vụ đặc biệt, nhưng những hệ thống trong danh mục này tập trung vào những tác vụ riêng biệt hơn, ít tổng quát. Có một nguyên tắc thô mà những công cụ soạn thảo biến đổi để đáp ứng cho những tác vụ đặc biệt hay những tình huống hướng dẫn có thể hỗ trợ tốt hơn cho những nhu cầu của người học và tác giả soạn thảo cho những tình huống này. Những hệ thống trong danh mục này được thiết kế bằng cách bắt đầu bằng một thiết kế hệ thống dạy học thông minh đặc biệt và tổng quát để tạo ra khung nền cho việc xây dựng những hệ thống dạy học tương tự. Việc soạn thảo tựa mẫu nhiều hơn trong những danh mục của những công cụ soạn thảo khác. Những tác giả soạn thảo thường được cho những ví dụ nguyên mẫu để giúp họ điền đầy vào khoảng trống. Một vấn đề tiềm năng với việc soạn thảo với mục đích đặc biệt là một khi một tác vụ và hướng tiếp cận hướng dẫn của nó được hệ thống hoá đủ để trở thành một mẫu, sau đó suy ra hệ thống phản ánh một hướng tiếp cận rất chuyên biệt cho việc biểu diễn và dạy học tác vụ đó.

Bởi vì chỉ những thứ mà hệ thống trong danh mục này có theo cách chung cho rằng chúng hỗ trợ những loại đặc biệt những tác vụ trong lĩnh vực, ta không nói bất kỳ thứ gì mà tổng quát về những loại hệ thống dạy học được xây dựng hay về sự thử nghiệm người học sử dụng hệ thống dạy học.

#### **2.6.1.7/ Chương trình dạy học dựa trên media thích ứng / thông minh :**

Bởi vì những hệ thống dựa trên phương tiện đa truyền thông và những công cụ soạn thảo dựa trên Web trở nên tinh vi hơn, chúng tăng việc hợp tác những phương thức và những mô hình từ lĩnh vực dạy học thông minh. Từ khi những hệ thống này và những công cụ soạn thảo của chúng trở nên nổi bật hơn, ta tạo ra một danh mục phân chia chúng. Chức năng của những hệ thống này chồng lên những chức năng từ danh mục 1 và 2 (phụ thuộc vào sự tập trung hướng dẫn ở mức hoặc là vĩ mô hoặc là vi mô). Đây là trường hợp với hầu hết những hệ thống dựa Web ngày nay. Mức tương tác và tính đúng đắn cho người học là thấp cho những hệ thống dạy học được xây dựng với những công cụ soạn thảo này. Không giống như những hệ thống trong danh mục khác, những hệ thống này phải quản lý những siêu liên kết giữa những đơn vị của nội dung (cũng như sự hình thành và sắp thứ tự nội dung). Những sự liên kết cho người học có thể được lọc, sắp xếp và chú giải một cách thông minh dựa vào mô hình người học hay tập hồ sơ. Việc lọc liên kết có thể được căn cứ vào những điều kiện tiên quyết, việc tải nhận thức, sự tương thích chủ đề, độ khó, ...

#### **2.6.2/ Soạn thảo nội dung của một ITS :**

Nếu như mô tả những ITSAT trong những mục nền tảng của những loại hệ thống dạy học mà chúng ta có thể xây dựng ta sẽ đi vào mô tả những tính năng của chính những công cụ soạn thảo. ITS thường được mô tả gồm có 4 thành phần chính : giao diện người học, mô hình lĩnh vực, mô hình dạy học, mô hình người học. Mặc dù những phần này không luôn luôn đủ để mô tả một ITS, nhưng tính năng của những

ITSAT có thể được mô tả tốt nhất bằng những mục của việc soạn thảo 4 thành phần này.

#### **2.6.2.1/ Soạn thảo giao diện :**

Thiết kế giao diện là một lĩnh vực nơi mà những công cụ soạn thảo multimedia truyền thống làm tốt hơn những ITSAT. Đây có thể bởi vì tạo ra một bộ xây dựng giao diện là khá tốn thời gian. Từ khi soạn thảo đồ họa cơ bản là một vấn đề được giải quyết thì hầu hết những nhà nghiên cứu soạn thảo ITS đã không ưu tiên nỗ lực để tạo ra những công cụ xây dựng đồ họa đầy đủ. Tuy nhiên, kinh nghiệm nghiên cứu cho thấy việc tùy biến giao diện dạy học là một quyền ưu tiên cho tác giả soạn thảo. Cũng vậy, việc ràng buộc giao diện người học đến những màn hình và cách trình bày đã được định nghĩa trước sẽ ràng buộc một cách khắt khe những loại tác vụ và sự tương tác mà một ITS có thể có với người học.

Ba hệ thống soạn thảo RIDES, SIMQUEST và Eon cho phép tác giả soạn thảo xây dựng giao diện hệ thống dạy học hoàn toàn từ ban đầu, sử dụng những đối tượng giao diện như nút nhấn, văn bản, các slide, hình ảnh, phim và những đối tượng ở mức thấp. Những đối tượng giao diện trong hệ thống này là “sống động”, trong đó chúng có thể viết kịch bản để phản hồi đến người dùng và những sự kiện được phát sinh trong chương trình, và những thuộc tính của chúng (màu sắc, vị trí, ...) có thể được thiết lập phụ thuộc vào nhiều giá trị khác nhau trong hệ thống dạy học. Với hệ thống RIDES, tác giả soạn thảo có thể định nghĩa những thành phần chính, thành phần phụ, và những liên kết vật lý. Trong hệ thống Eon, tác giả soạn thảo định nghĩa những mẫu màn hình đồ họa và hệ thống tự động tạo ra một cơ sở dữ liệu cho việc nắm giữ nội dung mẫu. Chẳng hạn nếu một màn hình chứa một đoạn phim, một câu hỏi và một lời giải thích được soạn thảo, tác giả soạn thảo có thể sử dụng một công cụ đầu vào dữ liệu để dễ dàng điền vào tên văn bản và tên đoạn phim cho hàng tá những màn hình tương tác này.

Những tính năng hỗ trợ một cách chủ động cho tác giả soạn thảo trong việc thiết kế một giao diện ITS, chẳng hạn bằng cách phân tích thiết kế giao diện cho rõ ràng và dễ sử dụng, chưa bao gồm trong những hệ thống soạn thảo ITS. Phần lớn những hệ thống soạn thảo bảo đảm những thiết kế giao diện đơn giản bằng cách định nghĩa trước những giao diện người học, chứ không phải cung cấp những tính năng thiết kế giao diện tí nào.

Soạn thảo giao diện, mặc dù linh hoạt hơn, có những hiệu quả bên lề phủ nhận việc giải phóng tác giả soạn thảo khỏi việc thiết kế những giao diện nghèo nàn, và thêm vào những danh sách kĩ năng mà một tác giả soạn thảo phải có.

Những hệ thống trong danh mục 7 giảm tải công việc hiển thị giao diện trình duyệt Web. Mặc dù không việc soạn thảo giao diện nào được cho phép (hay được cần) cho những hệ thống này, nhưng khả năng trình bày của những trình duyệt Web làm cho dễ dàng để phát sinh những trang web với một độ hợp lý của sự tương thích với một nỗ lực rất nhỏ.

#### **2.6.2.2/ Soạn thảo mô hình lĩnh vực :**

Những ITS bao gồm sự biểu diễn của tri thức lĩnh vực, những mô hình giả lập, và chuyên môn giải quyết vấn đề. Những công cụ soạn thảo được xây dựng cho mỗi một danh mục mô hình lĩnh vực này.

##### **2.6.2.2.1/ Những mô hình của tri thức và cấu trúc lĩnh vực :**

Một vài hệ thống soạn thảo bao gồm những công cụ cho việc hình ảnh hoá và soạn thảo nội dung những mạng các đối tượng (IDE, Eon, RIDES, và CREAM-Tools). Những công cụ này giúp cho tác giả soạn thảo hình ảnh hoá mối quan hệ giữa những thành phần lĩnh vực (như chủ đề, khoá học, những khái niệm và những thủ tục) và cho phép một cái nhìn sắc bén của vấn đề môn học. Một vài công cụ bị giới hạn cho sự biểu diễn theo cấu trúc cây của những khoá học, module, chủ đề ..., nhưng hầu hết cho

phép sự biểu diễn mạng được tạo hình tự do hơn. Sự hiển thị của các nút trong sự biểu diễn dạng cây là tự động, trong khi đối với sự biểu diễn mạng các tác giả soạn thảo phải định vị các nút đó.

Tri thức lĩnh vực có thể bao gồm tri thức về những thuộc tính liên quan đến dạy học về những chủ đề, chẳng hạn tầm quan trọng và độ khó của nó. Hầu hết những công cụ soạn thảo trong danh mục 1, 2, 5 đều bao gồm khả năng để soạn thảo những thuộc tính chủ đề. Một vài hệ thống (IDE, IRIS và CREAM-Tools) cung cấp những công cụ cho việc soạn thảo các mục đích hướng dẫn được chia theo chủ đề.

#### **2.6.2.2.2/ Sự giả lập và những mô hình :**

RIDES và SIMQUEST bao gồm những công cụ WYSIWYG tinh vi cho việc xây dựng những mô hình thiết bị và hiện tượng vật lý khác. Trong RIDES, các tác giả soạn thảo tạo ra những thành phần nhỏ, chẳng hạn nút bấm, nút chuyển đổi, những đòn bẩy, các ống, những hộp đen điện tử ... cùng với thuộc tính của chúng (như màu sắc, tình trạng tắt/mở ...)

#### **2.6.2.2.3/ Những mô hình của chuyên môn lĩnh vực :**

Chuyên môn lĩnh vực có thể bao gồm một vài loại tri thức, bao gồm chuyên môn giải quyết vấn đề, những kỹ năng thủ tục, những khái niệm và những sự kiện.

Những hệ thống soạn thảo trong danh mục 1, 2, 5 cho phép những tác giả soạn thảo biểu diễn những sự kiện, mối quan hệ và thủ tục đơn giản. Những sự kiện và mối quan hệ được lưu trữ như sự kết hợp (ví dụ: màu của X là Y, A chữ viết hoa của B). Những thủ tục đơn giản được lưu như sự sắp xếp thứ tự các bước, và một vài hệ thống có khả năng soạn thảo các thủ tục con. Hầu hết hệ thống sử dụng mạng nội dung kết hợp những thông tin lĩnh vực vào trong mô hình cấu trúc lĩnh vực. Chẳng hạn, một mạng chủ đề có thể liên kết những khái niệm đến những khái niệm con (với liên kết

“Một-là”) và những thủ tục chính đến thủ tục con. Loại thông tin này là cả nội dung và các đặc điểm kỹ thuật lĩnh vực.

Chuyên môn thủ tục cho sự vận hành thiết bị và những thủ tục khác được biểu diễn sử dụng sự biểu diễn tựa kịch bản đơn giản với những bước chính, các bước con, và những nhánh quyết định giới hạn. Nhiều hơn những kỹ năng phức tạp và những kỹ năng giải quyết vấn đề đòi hỏi sự biểu diễn dựa quy tắc sản xuất được sử dụng trong hệ chuyên gia (hay một hình thức phức tạp tương tự, ví dụ sự ràng buộc).

Như trường hợp những hệ thống công cụ soạn thảo khác, việc soạn thảo những sự kiện, mối quan hệ và những thủ tục đơn giản là tương đối dễ dàng. Đối với chuyên môn lĩnh vực được mô hình hoá với những quy tắc, ngữ pháp, hay sự ràng buộc, việc soạn thảo yêu cầu nhiều hơn. Những công cụ soạn thảo cho những loại sự biểu diễn này không được trình bày để sử dụng bởi những người không biết lập trình.

#### **2.6.2.2.4/ Loại tri thức lĩnh vực :**

Như đề cập ở trên, những hệ thống trong danh mục 5 phân biệt những loại tri thức khác nhau và có những sơ đồ biểu diễn tri thức khác nhau và có những chiến lược dạy học khác nhau cho mỗi loại. Cấu trúc này hướng dẫn và ràng buộc việc soạn thảo. Ví dụ Hệ thống DNA được sử dụng để soạn thảo các tri thức sự kiện, khái niệm và thủ tục. Những loại tri thức này có liên quan bởi liên kết “cái gì, như thế nào và tại sao”. Chẳng hạn, một tác giả soạn thảo tạo ra một đơn vị môn học cho “độ lệch tiêu chuẩn” được gợi ý để tạo ra nội dung thêm miêu tả “như thế nào để tính” (thủ tục), “tại sao” nó quan trọng (khái niệm) và “cái gì được sử dụng” (sự kiện) ...

#### **2.6.2.3/ Soạn thảo mô hình dạy học :**

Các chiến lược dạy học định rõ nội dung được sắp thự tự như thế nào, loại phản hồi nào được đưa ra, khi nào và như thế nào để dạy, giải thích, sửa sai, tóm lược, đưa ra một vấn đề v.v... Một sự thay đổi những phương pháp biểu diễn được sử dụng để

mô hình hoá chuyên môn dạy học bao gồm những thủ tục, kế hoạch, các ràng buộc và các quy tắc. Tuy nhiên, đa số ITSAT bao gồm một mô hình dạy học sửa lỗi.

#### **2.6.2.3.1/ Những hệ thống dựa kế hoạch :**

Một vài hệ thống (IRIS, GTE, IDE, và REDEEM) bao gồm cơ chế dựa trên kế hoạch với sự biểu diễn của cấu trúc cây đa mức của các mục đích, chiến lược và nhiệm vụ hướng dẫn. Chẳng hạn, khung nền IRIS bao gồm 3 mức: tiến trình nhận thức, những sự kiện hướng dẫn và các hành động hướng dẫn. IDE là duy nhất trong những bộ soạn thảo định rõ mỗi quy tắc lên kế hoạch dựa trên lý trí, để mỗi quy tắc có thể được “bào chữa” bởi một lý thuyết dạy học riêng biệt. Những quy tắc lên kế hoạch trong một vài hệ thống được sửa nhưng IDE và GTE cho phép những tác giả soạn thảo ghi trong những quy tắc lên kế hoạch mà định nghĩa một cấu trúc cây của những tác vụ con. Chẳng hạn : “Đề dạy những chức năng gồm các bước → 1 : Hiện thị chức năng; 2 : Dạy những tiến trình liên kết; 3 : Dạy những chức năng phụ; 4 : Hiện thị tóm lược”. Mục “Dạy những tiến trình liên kết” có thể được định nghĩa xa hơn sử dụng quy tắc khác. Tuy nhiên, những công cụ hình ảnh hoá và soạn thảo được cung cấp cho những hệ thống như là tối thiểu và tác vụ soạn thảo yêu cầu sự lập trình quan trọng hay những kỹ năng có tính kỹ thuật về tri thức.

#### **2.6.2.3.2/ Đa chiến lược :**

Một vài hệ thống bao gồm đa chiến lược dạy học và chọn một cách động các chiến lược thích hợp dựa trên nội dung và bản chất người dùng. Những hệ thống trong danh mục 5 có một nhóm những chiến lược dạy học đơn giản tương ứng, một chiến lược cho mỗi loại tác vụ hay tri thức được nhận ra bởi hệ thống. Chẳng hạn, một chiến lược khác sẽ được sử dụng để dạy các sự kiện, thủ tục và khái niệm. Không có sự soạn thảo chiến lược cho những hệ thống này. Những hệ thống REDEEM và Eon cho phép những tác giả soạn thảo định nghĩa đa chiến lược và “siêu chiến lược” cho việc lựa chọn một cách động giữa nhiều chiến lược.

#### **2.6.2.3.3/ Từ vựng cho hoạt động dạy học :**

Những hệ thống phát triển này trong danh mục 2 (và nhiều Vở ITS) có những từ vựng phức tạp được phát triển cho việc mô tả những phương pháp hướng dẫn. Những chiến lược hay quy tắc dạy học sau đó được sử dụng để xác định loại hành động cần tạo bất kỳ thời điểm nào đưa ra. Những hành động dạy học ví dụ bao gồm lời gợi ý, lời giải thích, sự sửa sai, tóm lược, thực hành, lựa chọn một chủ đề, và phản ánh dựa vào bài tập. Hầu hết những hệ thống như vậy có một lớp từ vựng trong đó một vài hành động được mở rộng vào trong hành động khác. Chẳng hạn, hệ thống GTE có vài trăm mục trong thư viện những tác vụ và phương pháp hướng dẫn của nó.

#### **2.6.2.4/ Soạn thảo mô hình người học :**

Hầu hết những hệ thống được đề cập trong phần này đều sử dụng những mô hình người học chồng lấp; chẳng hạn những chủ đề hay những bước thủ tục được chỉ định một giá trị dựa vào sự thực hiện của người học. XAIDA và Eon cho phép tác giả soạn thảo định nghĩa những khái niệm khó hiểu cũng như những chủ đề để cho hệ thống dạy học có thể đánh giá và sửa sai những lỗi chung. Demonstr8 dường như là hệ thống suy nhất sử dụng mô hình người học có thể chạy. Eon dường như là hệ thống duy nhất cho phép mô hình người học được soạn thảo.

#### **2.6.3/ Những phương pháp soạn thảo và phương pháp đạt được tri thức :**

Kế tiếp ta sẽ thảo luận những phương pháp tổng quát được sử dụng bởi những hệ thống soạn thảo để làm đơn giản và tự động việc soạn thảo và sự đạt được tri thức. Những phương pháp này là tổng quát, trong đó chúng có thể được sử dụng để cải thiện việc soạn thảo cho bất kỳ phần nào của 4 phần chính của một ITS được miêu tả ở trên, và có thể được sử dụng trong một công cụ soạn thảo cho bất kỳ công cụ nào trong 7 danh mục của những công cụ soạn thảo đã được miêu tả.

Bao gồm có tất cả 8 phương pháp như sau :

- (i) Làm khớp tri thức với những mô hình.



- (ii) Tri thức nhúng và tri thức mặc định
- (iii) Quản lý tri thức
- (iv) Hình ảnh hoá tri thức
- (v) Sự luận ra tri thức và quản lý dòng công việc
- (vi) Xác định tính hợp lý của thiết kế và tri thức
- (vii) Tái sử dụng tri thức
- (viii) Việc tạo tri thức tự động

### **2.7/ Kết luận :**

Việc xây dựng và sử dụng những công cụ để hỗ trợ trong việc phát triển ITS là đầy những khó khăn không thể lường hết. Những sự lựa chọn không may trong tiến trình thiết kế ITSAT có thể giới hạn tính linh hoạt và hữu dụng của những ITS kết quả. Những nhà phát triển ITSAT phải ý thức về những phiên bản phát triển ITS cũng như những phiên bản của ITSAT.

Như được mô tả trong chương này, có nhiều sự song song mà có thể vẽ ra giữa những ITSAT và công việc đang được làm trong lĩnh vực ESS. Cả hai đều có cùng mục đích và trải qua rất nhiều vấn đề giống nhau. Những phiên bản của việc sử dụng lại những thành phần, kiểm soát hệ thống và sự phát triển của những giao diện lĩnh vực đặc biệt liên quan đến những vùng lĩnh vực này và sự xem xét những phiên bản này là rất quan trọng.

Cũng vậy, khi cố gắng xây dựng một thứ gì đó có bản chất tổng quát, vấn đề của tính tổng quát chống lại sự chuyên biệt hoá xuất hiện. Trong công việc hiện tại, một điểm trung gian giữa hai hướng tiếp cận này đã được chấp nhận khi engine dạy học và những chiến lược dạy học của một ITS đã được giữ tổng quát trong sự chuyên biệt hoá được khuyến khích trong những lĩnh vực của sự định nghĩa lĩnh vực và sự phát triển

của những giao diện người dùng. Điều này có thể cung cấp một môi trường linh động mà vẫn còn có sức mạnh để mô hình hoá một cách thích hợp một loạt những lĩnh vực.

Việc dạy học những kỹ năng thủ tục là lĩnh vực nghiên cứu có giá trị cho việc phát triển những ITS khi rất nhiều những môi trường thế giới thực mới mà sự dạy học như vậy thì liên quan một cách đặc biệt. Những môi trường được giả lập cho những lĩnh vực đắt đỏ về vật liệu hay nguy hiểm có thể được định nghĩa và kiểm tra những kỹ năng đặc biệt. Sử dụng sự giới hạn như những giả định thế giới tĩnh cho phép những sự giới hạn để được đặt vào sự phát triển hệ thống nhưng không ảnh hưởng bất lợi đến môi trường học tập.

Một sự phân loại các công cụ soạn thảo là điều cần thiết để ta có cái nhìn tổng quan về những hệ thống đã phát triển. Từ đó, ta có thể định hướng tốt hơn cho những hệ thống sẽ được phát triển trong tương lai, cũng như sử dụng hiệu quả các công cụ hiện có cho việc tạo ra những ITS theo mong muốn sao cho ít tốn kém nhất bằng cách sử dụng những phương thức soạn thảo để soạn thảo nội dung cơ bản nhất của một ITS cần có đã được giới thiệu.

## **Chương 3 : Phát triển ITS sử dụng Cấu trúc tri thức**

### **3.1/ Giới thiệu :**

Một trong những lý do mà ITS khó có thể được xây dựng hay tốn nhiều thời gian và tiền bạc để xây dựng là lĩnh vực được chọn để thực hiện. Thật không cần thiết để khái niệm hoá hoàn chỉnh toàn bộ, vì điều này không đơn giản để đạt được. Hệ thống không chỉ yêu cầu thông tin về loại vấn đề được trình bày cho người học, về cách mà chúng giải quyết, nhưng nó cũng phải biết về những lỗi tiềm năng mà người học có thể mắc phải và sự can thiệp có thể ở một mức nào đó để hướng dẫn sửa sai những vấn đề này. Cho đến nay, những lĩnh vực có độ phức tạp không cao đã được khái niệm hoá.

Phương pháp dùng để cài đặt tri thức lĩnh vực là điều cũng cần xem xét. Vì mỗi lĩnh vực khác nhau có thể có những mô hình tri thức khác nhau, độ phức tạp cũng khác nhau. Làm sao để tri thức phải được mã hoá một cách rõ ràng, dễ hiểu và có thể tiên hoá là điều không dễ dàng. Trong đề án này, ta sẽ mô hình hoá tri thức lĩnh vực bằng các **Cấu trúc tri thức**.

Một trong những mục đích của hệ thống mà khóa luận muốn xây dựng là phát triển một công cụ cho những giáo viên ( không cần có kinh nghiệm về lập trình ) cũng có thể thảo ra các bài tập giảng dạy trên nhiều lĩnh vực khác nhau. Giáo viên sẽ sử dụng tri thức đã được mã hoá để chọn ra những nội dung theo yêu cầu dạy cho để hệ thống tự động phát sinh bài tập. Ngoài ra, hệ thống còn được xây dựng thêm một bộ giả lập để trẻ có thể làm bài tập trên đó mà giáo viên đã thảo ra trước đó. Một bộ hướng dẫn sẽ được xây dựng tiếp theo để có thể hướng dẫn thông minh cho trẻ giải quyết vấn đề mà chúng gặp phải trong quá trình làm bài tập. Và trong tương lai có thể xây dựng thêm một bộ đánh giá trẻ để hoàn thiện toàn bộ một hệ thống ITS hoàn chỉnh.

Trong phần sau (Mục 2) phạm vi lĩnh vực được mô tả. Trong Mục 3, ta sẽ trình bày phương pháp luận để khái niệm hoá tri thức lĩnh vực. Những cấu trúc tri thức được tổng hợp từ lĩnh vực được mô tả trong Mục 4. Tiếp đó những đối tượng chủ yếu sử dụng trong tri thức được tổng hợp trong Mục 5. Trong Mục 6, ta sẽ nghiên cứu những người sử dụng hệ thống. Kiến trúc cơ bản của hệ thống được xây dựng sẽ minh hoạ trong Mục 7. Trong mục 8, ta diễn tả những tiến trình cơ bản trong hệ thống. Phần cuối ta sẽ tóm lại những gì đã được xây dựng và những gì sẽ được xây dựng tiếp trong tương lai.

### 3.2/ Phạm vi lĩnh vực :

Trẻ trong những năm đầu phát triển là nhóm người dùng có lợi từ những kĩ thuật học tập này. Chúng phải học nhiều để biết và hiểu được các mối quan hệ tồn tại trong thế giới mà chúng sống. Một vài lĩnh vực cần đào tạo và rèn luyện là, giữa nhiều lĩnh vực khác nhau, đọc và viết, và những ý niệm về toán và âm nhạc.

Một điều cần phải tính đến cho việc áp dụng mô hình vào lĩnh vực cần thực hiện quy cho giả định rằng một hệ thống hỗ trợ dạy học là phương tiện để rèn luyện những kĩ năng đã được dạy cho trẻ trước đó bởi giáo viên hoặc những người thân xung quanh (cha mẹ, anh em, bạn bè) , hay bởi mối quan hệ của chúng với môi trường xã hội (tivi, báo chí, ...). Một khi những khái niệm này được hiện diện trong trẻ, hệ thống hỗ trợ dạy học sẽ dễ dàng xây dựng "*một ý nghĩa bên trong của các khái niệm*" bằng cách trình bày những hoạt động khác nhau cho những đối tượng và mối quan hệ tương ứng.

Những ứng dụng của mô hình được chỉ rõ trong sự phát triển của các kĩ năng toán học cơ bản, đặc biệt là sự tạo nên khái niệm con số trong trẻ trước tuổi đến trường. Lĩnh vực này phù hợp với những yêu cầu về việc đào tạo và rèn luyện rộng rãi như là một phương pháp luận làm việc có thể. Hệ thống hỗ trợ dạy học cung cấp cho trẻ những bài tập rèn luyện khác nhau, được soạn bởi giáo viên.

Những trẻ trước tuổi đến trường (từ 2 đến 5 tuổi) đang trong quá trình phát triển nhiều kĩ năng về nhận thức, xúc cảm và xã hội. Trong lĩnh vực nhận thức là yêu cầu về

ngôn ngữ, sự phát triển của quá trình đọc viết và việc phán đoán toán học một cách logic. Quá trình này đi đôi với sự phát triển của các kỹ năng về trí tuệ. Tất cả những khả năng này sẽ được dùng để xây dựng những dạng khác nhau của tri thức, được học trong những điều kiện và hoàn cảnh khác nhau.

Rõ ràng là trẻ em có các bản tính riêng trong tuổi trước khi đến trường : thuộc tính bản năng, niềm vui thích lượm nhặt các đồ vật, niềm yêu thích công việc lặp đi lặp lại, sự thèm muốn xem xét kỹ lưỡng đồ vật, và thích sắp xếp các đồ vật. Điều này có thể được sử dụng cho việc thiết kế những bài tập có hướng dẫn.

### 3.3/ Phương pháp luận khái niệm hoá :

Một khi lĩnh vực được hình thành thì cần thiết phải *khái niệm hoá* nó. Những phương pháp luận sau được dùng :

+ **Bước 1** : Định nghĩa những nội dung và khái niệm có liên quan đến lĩnh vực của các kỹ năng toán học cơ bản.

Bắt đầu từ việc nghiên cứu được thực hiện sớm trên việc giáo dục toán học và tâm lý nhận thức, đặt biệt từ những công việc được phát triển bởi Jean Piaget, sự thoả ra khái niệm con số có liên quan đến sự phát triển của những kỹ năng nhất định mà có thể nhóm thành sự phân loại, sự tương đương, sự liên tiếp và sự bảo toàn số lượng. Những kỹ năng và khả năng này có lẽ được tổ chức trong các lĩnh vực công việc sau :

- Lĩnh vực ngôn ngữ : liên quan đến những bài tập về từ được học trong môi trường xã hội có nhiều ý nghĩa trong ngữ cảnh toán học. Lĩnh vực này bao gồm việc sử dụng đúng những từ chỉ số lượng (tất cả, một vài, nhiều, một ít, không có, ...); những từ so sánh (nhỏ, lớn, cao, thấp, đầy, rỗng, ...); từ vựng chỉ không gian (lên, xuống, bên trong, bên ngoài, phía trước, bên cạnh, trái, phải, ...); những dạng hình học (hình tròn, hình tam giác, hình vuông, hình lập phương, hình cầu, hình khối, ...); từ chỉ thời gian (trước, sau, ngày hôm qua, ngày hôm nay, ngày mai, ...) và những con số đầu tiên (một, hai, ba, ..., 10).

- Lĩnh vực tập hợp : một trong những khái niệm cơ bản trong việc học toán. Nó bao gồm những khái niệm “thuộc” hay “không thuộc” đối với một tập hợp cho trước, sự tương ứng giữa những đối tượng của hai tập hợp phụ thuộc vào mối quan hệ, sự tương ứng giữa hai hay nhiều tập hợp, và sự phân loại tùy theo một hay nhiều tiêu chuẩn (khái niệm của lớp đối tượng).

- Lĩnh vực tuần tự : bao gồm những thành phần được sắp xếp của một nhóm theo một tiêu chuẩn ( ví dụ : kích thước, màu sắc, số lượng, ...).

- Lĩnh vực biến đổi : liên quan đến điều kiện mà số lượng và số các đối tượng trong tập hợp không thay đổi trong một tập hợp không phụ thuộc vào sự sắp xếp vật lý hay sự xuất hiện bên ngoài cho những số lượng liên tục hay không liên tục.

+ **Bước 2** : Liệt kê các phương pháp luận mà có thể được sử dụng trong các bài tập có các khái niệm này.

Nhìn chung có thể mong rằng trẻ có thể thăm dò, thanh minh, trình bày, giải quyết, xây dựng, thảo luận, sử dụng, kiểm tra, mô tả, phát triển và dự đoán những mối quan hệ bằng cách sử dụng các khái niệm đã được đề cập trước đó. Để đạt mục đích này, chúng phải có cơ hội được tiếp xúc với số lượng các vật dụng khác nhau để thao tác (các thẻ bài, các khối hình hộp có thể lắp ráp, các que, các khối hộp logic, các đồ khảm, các thanh màu, các nút điều khiển, các hộp đựng, ...).

- Những nhiệm vụ được thực hiện trong lĩnh vực ngôn ngữ là :

- Nhận biết được những thuộc tính của các đối tượng để thao tác chúng.
- Thiết lập sự so sánh giữa các đối tượng.
- Xem xét kĩ sự giống nhau và khác nhau tùy theo khía cạnh chất lượng và số lượng.

- Trong lĩnh vực tập hợp, những nhiệm vụ cần thực hiện :

- Hình thành các tập hợp với những đối tượng cụ thể.
- Nhận biết mối quan hệ của việc thuộc hay không thuộc tập hợp
- Phân biệt và sử dụng các khái niệm: tập hợp, phần tử, thuộc.

- Phân biệt và đặt tên các tập hợp tương ứng.
  - Phân biệt và đặt tên các tập hợp rỗng.
  - Nhận biết và xác định số các phần tử của một tập hợp.
  - Phân biệt và sử dụng các khía niệm về số lượng (tất cả, nhiều, một vài,...).
  - Phân loại các đối tượng dựa vào đặc tính sử dụng của chúng (đối tượng được sử dụng để làm gì ?).
  - Phân loại các đối tượng dựa vào một vài tiêu chuẩn khác.
  - Phân loại vật dụng có cấu trúc sử dụng mỗi tiêu chuẩn một lần: màu sắc, hình dạng, hay kích thước.
  - Phân loại vật dụng có cấu trúc sử dụng 2 hay nhiều tiêu chuẩn đồng thời.
  - Thiết lập sự tương ứng giữa đối tượng và đối tượng dựa vào “Sự chèn vào“ (ví dụ: ráp vừa hình ảnh vào các đường rãnh).
  - Thiết lập sự tương ứng giữa đối tượng và đối tượng dựa vào sự giống nhau về cấu trúc tự nhiên (ví dụ: ly và đĩa ...).
  - Thiết lập sự tương ứng giữa đối tượng và dấu hiệu nhận biết (ví dụ: một người và tên của người đó)
  - Thiết lập sự tương ứng giữa dấu hiệu nhận biết và ký hiệu (ví dụ: trái tim và ♥).
- Trong lĩnh vực tuần tự, những nhiệm vụ bao gồm :
- Sắp xếp dãy liên tục của 3, 4, 5 hay 6 phần tử, từ nhỏ đến lớn và ngược lại.
  - Thiết lập sự tương ứng giữa hai dãy liên tục có thứ tự.
  - Thiết lập sự tương ứng chéo giữa hai dãy liên tục.
  - Cho biết vị trí của một phần tử trong một dãy tuần tự.

- Trong lĩnh vực biến đổi, trẻ phải thực thi những nhiệm vụ sau :
  - Nhận biết khi nào thì duy trì số lượng không liên tục.
  - Nhận biết khi nào thì duy trì số lượng liên tục.

+ **Bước 3** : Phân tích tính năng của từng loại phương pháp luận

Từ tất cả những loại phương pháp luận dạy học, cần thiết phải lựa chọn những phương pháp luận phù hợp để thực hiện trên máy tính. Máy tính chỉ là một công cụ, do đó, nó không thể thay thế việc sử dụng các thành phần cung cấp thông tin khác. Trẻ cần tiếp xúc và trải nghiệm với các đối tượng cơ bản mà những nhà sư phạm và những người thân xung quanh khác có trách nhiệm cung cấp cho chúng những vật dụng đó.

### 3.4/ Cấu trúc Tri thức :

#### 3.4.1/ Định nghĩa :

Điều cốt lõi của một hệ thống ITS là những quy luật mà nó lưu trữ. Trong khóa luận này, những quy luật này được lưu trữ dưới dạng Cấu trúc Tri thức, do đó một khi ta chọn được lĩnh vực cần thực hiện, ta có thể tổng hợp nó thành các Cấu trúc Tri thức.

*Vậy Cấu trúc Tri thức là một khái niệm hóa tri thức của một phần của lĩnh vực cần thực hiện và xác định tính năng cơ bản của một hoạt động.*

#### 3.4.2/ Tính chất :

Cấu trúc tri thức thực chất là một phần của tri thức lĩnh vực đã được khái niệm hóa sao cho nó tổng quát hơn so với những tri thức thô. Nói cách khác, Cấu trúc tri thức là phần cơ sở đã được tổng hợp từ nhiều tri thức khác nhau của lĩnh vực nhưng có những thuộc tính và tính năng tương tự giống. Do đó, nếu tập hợp các cấu trúc tri thức với nhau ta sẽ có bộ tri thức của lĩnh vực đã được khái niệm hóa. Để phân biệt các Cấu trúc tri thức khác nhau, ta có thể dựa vào những đặc tính chung như sau :



+ Mỗi cấu trúc tri thức sẽ có một tính năng riêng xác định hoạt động của nó dùng để làm gì ? → Mục đích.

+ Mỗi cấu trúc tri thức sẽ có cách thức hoạt động khác nhau để thực hiện tính năng của nó. → Cách thức hoạt động.

+ Mỗi cấu trúc tri thức sẽ có những thông số riêng của nó dùng để điều khiển sự hoạt động. → Thông số.

### 3.4.3/ Các Cấu trúc tri thức :

Nhà sư phạm sẽ sử dụng những Cấu trúc Tri thức này để xây dựng nên những bài tập. Những bài tập này được xác định bằng cách chỉ rõ các đối tượng và các thông số của Cấu trúc Tri thức. Để định nghĩa lĩnh vực đã chọn , 6 Cấu trúc Tri thức sau được xác định là :

#### 3.4.3.1/ Cấu trúc Tri thức 1 :

Loại trừ những đối tượng không thuộc về một danh mục xác định.

- Mục đích: theo một tiêu chuẩn được chọn trước.

- Một số khái niệm:

+ Danh mục cơ bản: là danh mục gồm các đối tượng có cùng một tiêu chuẩn (chẳng hạn: cùng là hình tròn, cùng là cá, cùng là chim ...).

+ Đối tượng khác: là các đối tượng khác với các đối tượng trong danh mục cơ bản, tức khác tiêu chuẩn.

+ Vùng Trình Diễn: là vùng hiển thị danh mục cơ bản và đối tượng khác.

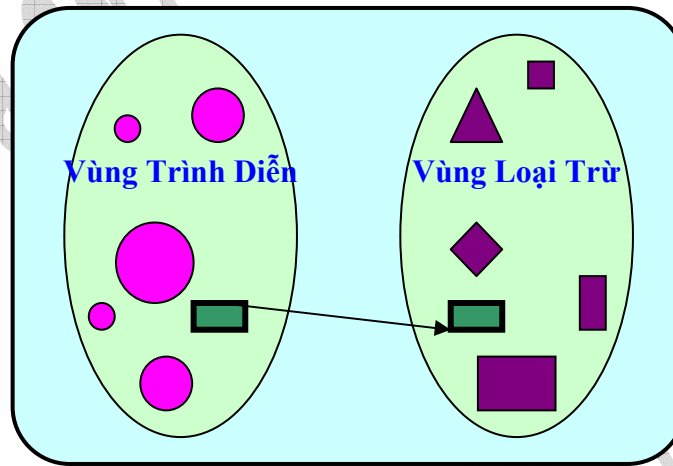
+ Vùng Loại Trừ: chứa các đối tượng khác, tức không phù hợp với tiêu chuẩn của danh mục cơ bản.

- Cách thức hoạt động: một nhóm hỗn lộn các đối tượng được hiển thị cho trẻ. Có rất nhiều đối tượng có cùng tiêu chuẩn (chúng thuộc về danh mục cơ bản), trong

khi những đối tượng còn lại thì khác. Những đối tượng phù hợp với tiêu chuẩn đã cho thì đưa về phía trái trong **Vùng Trình Diễn**, trong khi những đối tượng không phù hợp thì được di chuyển về **Vùng Loại Trừ**.

- Thông số: Tiêu chuẩn được chọn định nghĩa danh mục cơ bản. Những đối tượng thuộc về danh mục cơ bản hay danh mục khác.

- Ví dụ: Ở Hình 3.1, tất cả các đối tượng không phải là đường tròn phải được chọn ở **Vùng Trình Diễn** và di chuyển vào **Vùng Loại Trừ**. Một ví dụ thực tế là mang những động vật không sống dưới lòng đại dương lên bề mặt để chúng khỏi bị nghẹt thở ở dưới nước.



### H3.1 Theo một tiêu chuẩn được chọn trước : hình tròn

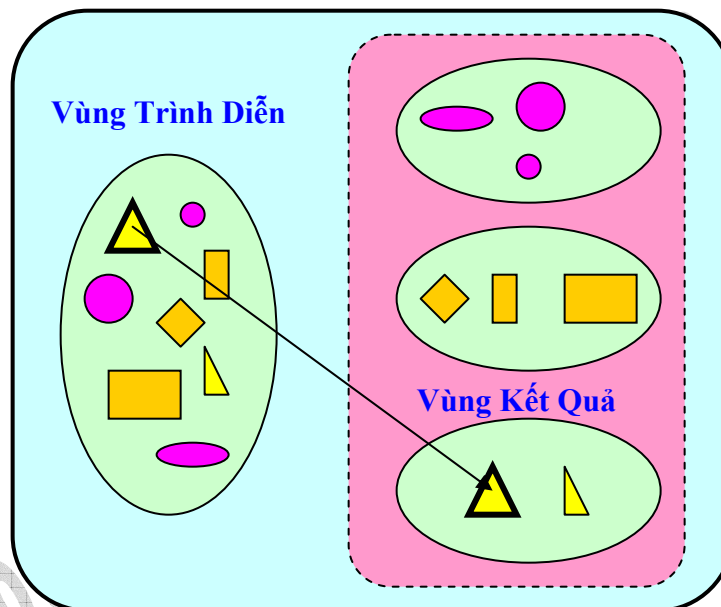
#### 3.4.3.2/ Cấu trúc Tri thức 2 :

Tổ chức các đối tượng để hình thành các danh mục.

- Mục đích: Xác định các tập con, được tổ chức một cách logic, từ một tập hợp cho trước.

- Một số khái niệm:

- + Danh mục được hiển thị: chứa các tập con các đối tượng có cùng tiêu chuẩn.
  - + Vùng Trình Diễn: vùng hiển thị danh mục đã tạo.
  - + Vùng Kết Quả: chứa các tập con của danh mục.
- Cách thức hoạt động: Một nhóm hỗn hợp các đối tượng thuộc về một vài danh mục được hiển thị cho trẻ trong **Vùng Trình Diễn**. Trong **Vùng Kết Quả**, một tập hợp các tập hợp con được hiển thị. Mỗi một danh mục được định nghĩa bởi một yếu tố được cho trước. Trẻ phải xác định các đối tượng thuộc **Vùng Trình Diễn** và định vị chúng trong các danh mục khác nhau thuộc **Vùng Kết Quả**.
- Thông số: Danh sách các mục để chọn. Những đối tượng cho từng danh mục.
  - Ví dụ: Ở Hình 3.2, trẻ sẽ chọn những hình tam giác, hình tứ giác, và hình tròn từ **Vùng Trình Diễn** và di chuyển chúng vào các tập con tương ứng trong **Vùng Kết Quả**. Một ví dụ thực tế của ví dụ tổng quát trên sử dụng các đối tượng trong thế giới của trẻ là sắp xếp các hàng tạp phẩm trong chợ tùy theo hình dạng, kích thước, hay loại thực phẩm ...



### H3.2 Tổ chức các đối tượng tạo thành các danh mục : hình tròn, tứ giác, tam giác

#### 3.4.3.3/ Cấu trúc Tri thức 3 :

Hình thành sự tuần tự.

- Mục đích: Sắp xếp tuần tự từ một tiêu chuẩn cho trước.

- Một số khái niệm:

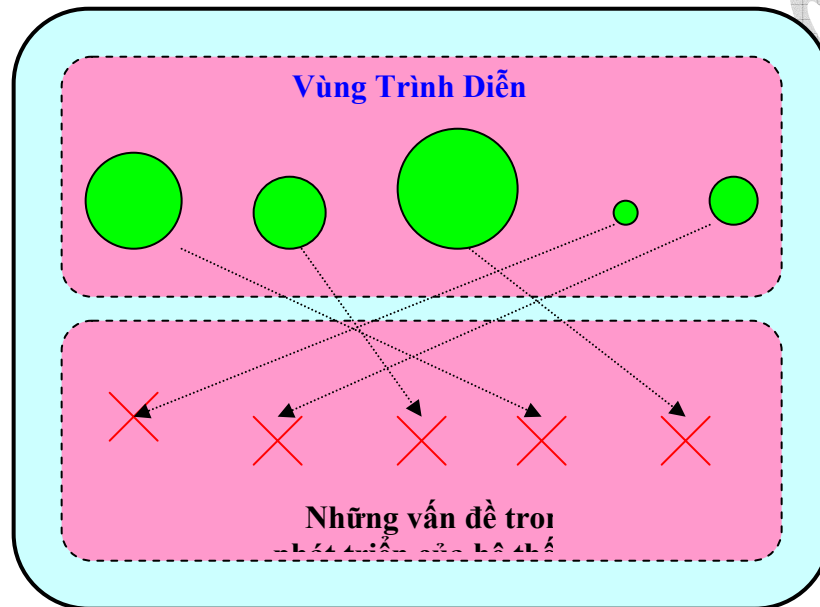
+ Vùng Trình Diễn: chứa các đối tượng (cùng tiêu chuẩn) nhưng khác nhau về kích thước hoặc giá trị... được đặt không theo thứ tự nào.

+ Vùng Tuần Tự: chứa các đối tượng đã được sắp xếp.

- Cách thức hoạt động: Một vài đối tượng, có thể được sắp trong một cách duy nhất tùy theo tiêu chuẩn đã cho, cùng với hướng của thứ tự (ví dụ từ lớn đến nhỏ) được hiển thị cho trẻ. Trẻ phải di chuyển những đối tượng từ **Vùng Trình Diễn** để hình thành nên dãy đã sắp thứ tự trong **Vùng Tuần Tự**.

- Thông số: Sự mở rộng của tuần tự; mức độ hỗn loạn của các đối tượng; cách để chọn những yếu tố tuần tự (ví dụ theo một, hai, hay nhiều yếu tố); danh sách các danh mục để chọn.

- Ví dụ: Ở Hình 3.3, những hình tròn với kích thước khác nhau được chọn từ **Vùng Trình Diễn** và di chuyển, tùy theo kích thước của chúng (từ nhỏ nhất đến lớn nhất) sang **Vùng Tuần Tự**. Một ví dụ thực tế là sắp thứ tự các bức tranh theo thứ tự thời gian để miêu tả quá trình rơi của chiếc bình hoa (chiếc bình hoa trên bàn, chiếc bình hoa đang rơi, chiếc bình hoa bể dưới sàn nhà).



### H3.3 Hình thành sự tuần tự của hình tròn từ nhỏ đến lớn

#### 3.4.3.4/ Cấu trúc Tri thức 4 :

Hoàn thành dãy tuần tự.

- Mục đích: Hoàn thành những mẫu có một hay hai chiều theo một tiêu chuẩn tuần tự cho trước.

- Một số khái niệm:

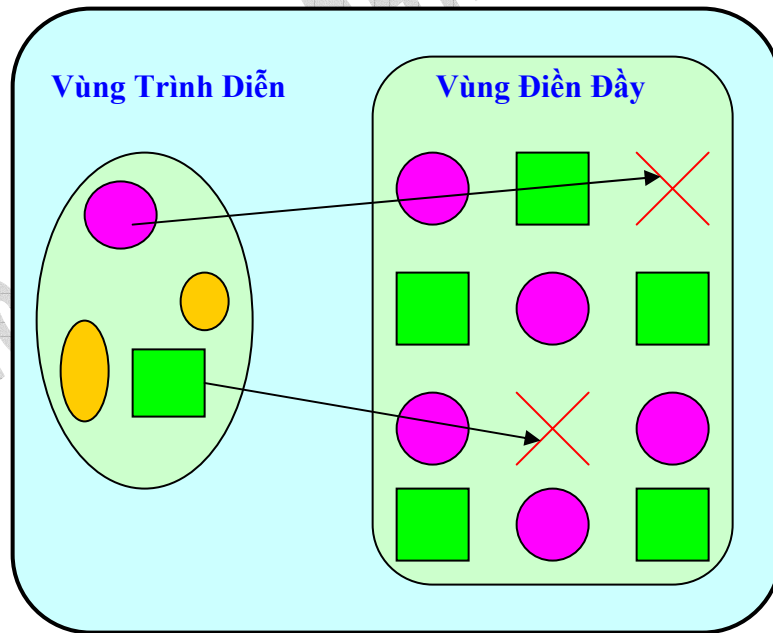
+ **Vùng Điền Đầy**: chứa các đối tượng được sắp xếp theo một sự tuần tự logic nào đó nhưng còn một hoặc nhiều đối tượng bị khuyết.

+ **Vùng Trình Diễn**: sẽ hiển thị nhiều đối tượng khác nhau, trong đó có các đối tượng giống với các đối tượng trong Vùng Điền Đầy.

- Cách thức hoạt động: trong **Vùng Điền Đầy**, một nhóm các đối tượng, theo một sự tuần tự có một hay hai chiều, được hiển thị nhưng có một hay hai đối tượng bị mất. Trẻ phải hoàn thành sự tuần tự bằng cách chọn các đối tượng từ **Vùng Trình Diễn** và định vị chúng một cách phù hợp trong **Vùng Điền Đầy**.

- Thông số: Loại và số lượng các đối tượng nguồn và đích. Mẫu đặt các đối tượng nguồn và đích. Số lượng các đối tượng bị mất.

- Ví dụ: Ở Hình 3.4, một vài hình vuông và hình tròn đã bị mất trong **Vùng Điền Đầy**, vì vậy trẻ phải chọn trong **Vùng Trình Diễn** các hình trên và đặt chúng vào vị trí phù hợp trong **Vùng Điền Đầy**. Một ví dụ trong thế giới thực là thay thế các sản phẩm trên kệ của siêu thị.



H3.4 Hoàn chỉnh sự tuần tự tròn vuông

#### 3.4.3.5/ Cấu trúc Tri thức 5 :

Thiết kế những sự liên kết chính xác.

- Mục đích: Thiết lập các mối quan hệ giữa các cặp đối tượng trong một lĩnh vực giới hạn được chọn.

- Một số khái niệm:

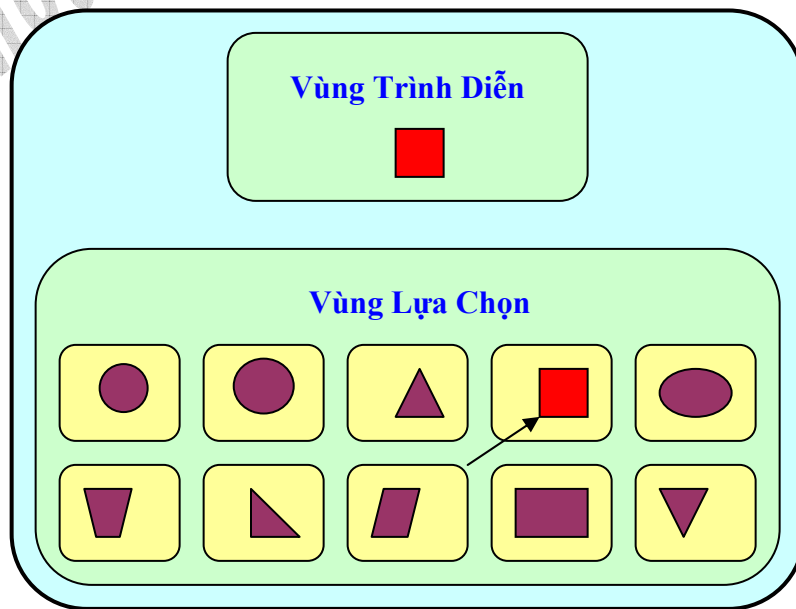
+ **Vùng Trình Diễn**: chứa một đối tượng được hiển thị tại một thời điểm nhất định.

+ **Vùng Lựa Chọn**: gồm nhiều đối tượng khác nhau để lựa chọn.

- Cách thức hoạt động: Một danh sách các đối tượng được hiển thị một lần tại một thời điểm trong **Vùng Trình Diễn**. Mỗi một đối tượng phải được liên kết với một trong những đối tượng trong **Vùng Lựa Chọn**, tùy theo tiêu chuẩn đã cho giữa chúng.

- Thông số: Những đối tượng tạo nên danh sách lựa chọn. Những đối tượng được hiển thị. Loại đối tượng của danh sách (hình ảnh hoặc âm thanh). Tiêu chuẩn mối quan hệ.

- Ví dụ: Ở Hình 3.5, trẻ phải nhận thấy hình vuông trong **Vùng Trình Diễn** và liên kết nó với đối tượng hình học tương ứng trong **Vùng Lựa Chọn**. Một ví dụ thực tế là chọn phù hợp bản chất âm thanh của một đối tượng với các hình ảnh khác nhau được hiển thị trong **Vùng Lựa Chọn**.

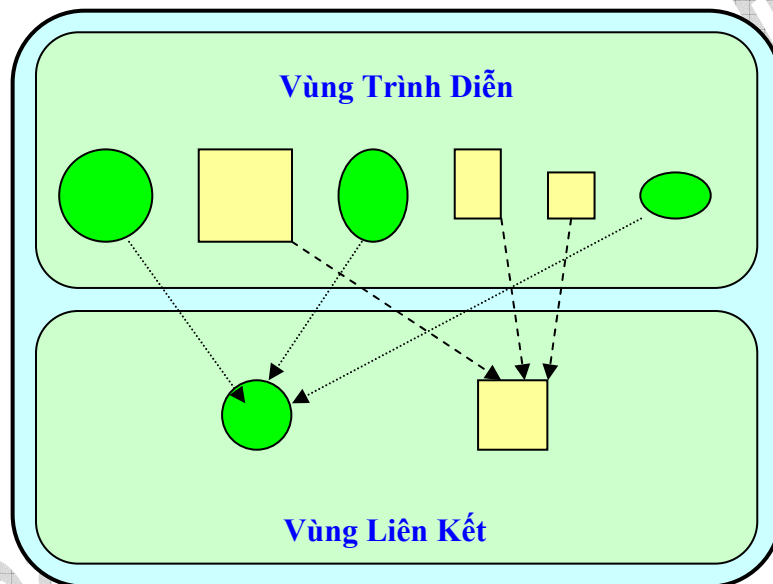


### H3.5 Thiết lập các mối liên kết chính xác, chẳng hạn hình vuông

#### 3.4.3.6/ Cấu trúc Tri thức 6 :

Thiết lập đa kết hợp.

- Mục đích: Thiết lập các mối quan hệ giữa cặp các đối tượng trong một phạm vi rộng.
- Một số khái niệm:
  - + Vùng Trình Diễn: chứa nhiều đối tượng được hiển thị đồng thời.
  - + Vùng Liên Kết: chứa nhiều đối tượng đã được hiển thị trước.
- Cách thức hoạt động: Nhiều đối tượng / câu hỏi được hiển thị đồng thời trong Vùng Trình Diễn, trẻ phải liên kết mỗi một chúng với những câu trả lời có thể khác nhau trong Vùng Liên Kết. Không phải tất cả các khả năng đều có câu trả lời.
- Thông số: Những đối tượng được liên kết. Số lượng các khả năng để chọn mỗi liên kết. Danh sách các danh mục để chọn các đối tượng. Tiêu chuẩn mỗi quan hệ.
- Ví dụ: Ở Hình 3.6, trẻ phải chọn những đối tượng tròn từ **Vùng Trình Diễn** và liên kết chúng với hình tròn của **Vùng Liên Kết**, thêm vào đó, trẻ phải chọn những hình tứ giác của **Vùng Trình Diễn** và nối chúng với hình vuông thuộc **Vùng Liên Kết**. Một ví dụ thực tế là trẻ thì mặc những bộ đồ nhỏ, còn người lớn thì mặc những bộ quần áo lớn.



H3.6 Thiết lập đa liên kết giữa đối tượng tròn và tứ giác



### **3.5/ Các đối tượng chính được thực hiện trên Cấu trúc tri thức :**

Cấu trúc tri thức sử dụng nhiều đối tượng khác nhau để xây dựng nên. Nhưng ta có thể tổng hợp được một số loại đối tượng chính mà các Cấu trúc tri thức trên sử dụng. Đó là 3 loại đối tượng sau:

#### **3.5.1/ Danh mục các đối tượng đơn giản :**

Bao gồm các tập hợp các đối tượng, thực chất là các hình ảnh, mà nội dung bằng hình xác định đối tượng được hiển thị (ví dụ: một con gà mái, một con gà trống, một con thỏ, một con vịt mái, một con ngỗng các, một con heo và một con ngựa tùy thuộc vào danh mục: những động vật nuôi ở nông trại), và sự mô tả của các đối tượng này thường tương ứng với tên của chúng ( ví dụ: “gà mái” ...).

#### **3.5.2/ Sự tuần tự của các đối tượng :**

Danh mục mà tập hợp nhóm của các đối tượng giống nhau về bản chất “sở hữu” một mối quan hệ thứ tự tùy theo một vài tiêu chuẩn về số lượng (ví dụ : sự sắp thứ tự của chai lọ từ kích thước nhỏ đến lớn). Thực chất của những đối tượng này được định hình bởi một hình ảnh, và một mô tả chung cho tất cả sự tuần tự (ví dụ: “Sự tuần tự của chai lọ theo kích thước”).

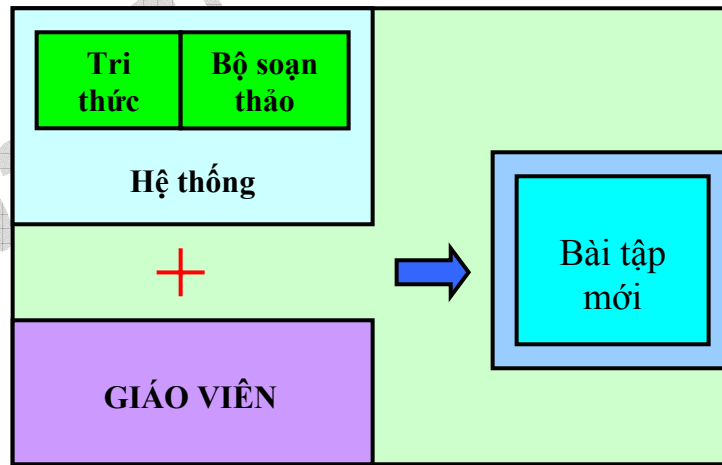
#### **3.5.3/ Cặp các đối tượng :**

Là sự kết hợp được hình thành bởi một hình ảnh liên kết với một hình ảnh hay một âm thanh khác (ví dụ: cặp hình ảnh – âm thanh: hình ảnh xe cứu thương liên kết với tiếng hú của xe; cặp hình ảnh – hình ảnh: hai bông hoa liên kết với số “2”).

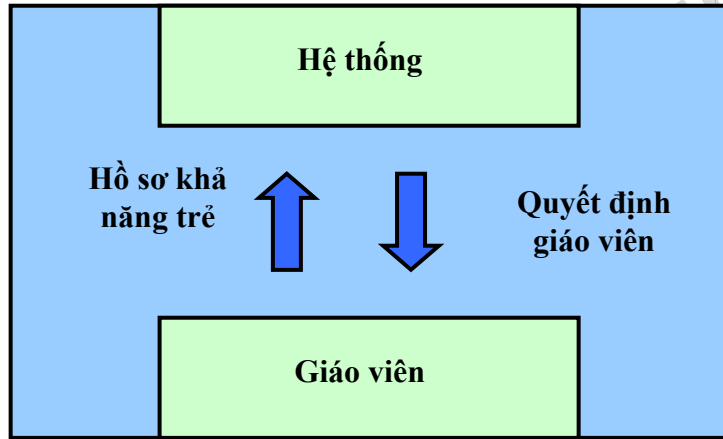
### **3.6/ Những người sử dụng của hệ thống :**

Ta có thể xác định được có 3 loại người sử dụng. Thứ nhất là trẻ, những người muốn thu thập thêm những kiến thức mới là kết quả của sự tương tác của chúng với hệ thống. Sự tham gia của trẻ được giới hạn vào những phần bài tập mà chúng đã chọn từ những dạng bài tập khác nhau.

Thứ hai là giáo viên, những người tham gia tích cực vào hệ thống và đóng 2 vai trò. Vai trò thứ nhất là không cần có những hiểu biết cần thiết về việc phát triển phần mềm nhưng họ chính là những người xây dựng các hoạt động bằng cách sử dụng hệ thống như một công cụ để phát sinh bài tập và quyết định nội dung bài tập từ những tri thức đã được lưu trữ (Hình 3.7). Vai trò thứ hai của người giáo viên là họ cũng chính là những người sử dụng bình thường (Hình 3.8). Hệ thống giảng dạy sẽ cho giáo viên biết về khả năng của trẻ trong và sau các lần làm bài tập.



H3.7: Giáo viên trong vai trò là “người xây dựng bài tập”



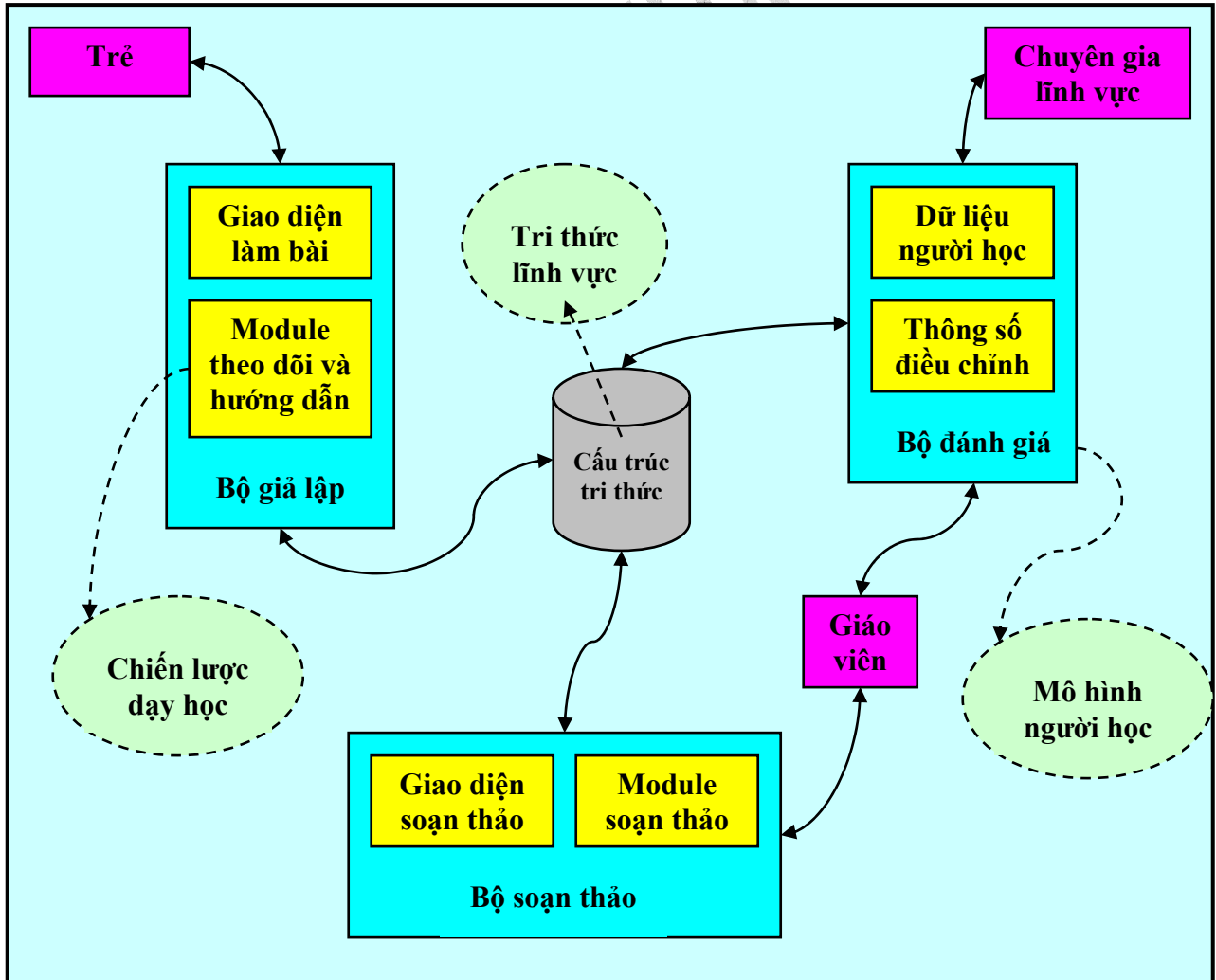
**H3.8: Giáo viên trong vai trò là “người sử dụng”**

Có một mối quan hệ rõ ràng giữa giáo viên và việc sử dụng phần mềm sư phạm trong trường học. Phần mềm được thiết kế cho phép giáo viên có thể thiết kế những mẫu bài tập riêng của họ. Giáo viên phải tưởng tượng ra lĩnh vực mà họ định dạy cho trẻ trong trường hợp những cấu trúc chung được hỗ trợ bởi phần mềm. Điều đó để nói lên giáo viên được khuyến khích trong việc khái niệm hóa cấp độ kiến thức mà họ dạy.

Loại người sử dụng thứ ba là những chuyên gia trong lĩnh vực này. Vai trò này được thể hiện qua những nhà sư phạm hoặc những nhà tâm lý về giáo dục - đó là những người biết về chiều sâu của mô hình cấp dưới của hệ thống. Họ có thể thay đổi chức năng bên trong của hệ thống sử dụng ngôn ngữ ở cấp độ cao bằng các thực đơn và đồ họa để tránh sự thay đổi cần thiết của mã nguồn gốc.

### 3.7/ Kiến trúc của hệ thống :

Dù kiến trúc của hệ thống ITS được xây dựng như thế nào thì nó cũng luôn gồm 4 thành phần cơ bản (chương 1). Hình 3.9 sẽ giới thiệu kiến trúc của hệ thống mà khóa luận xây dựng.

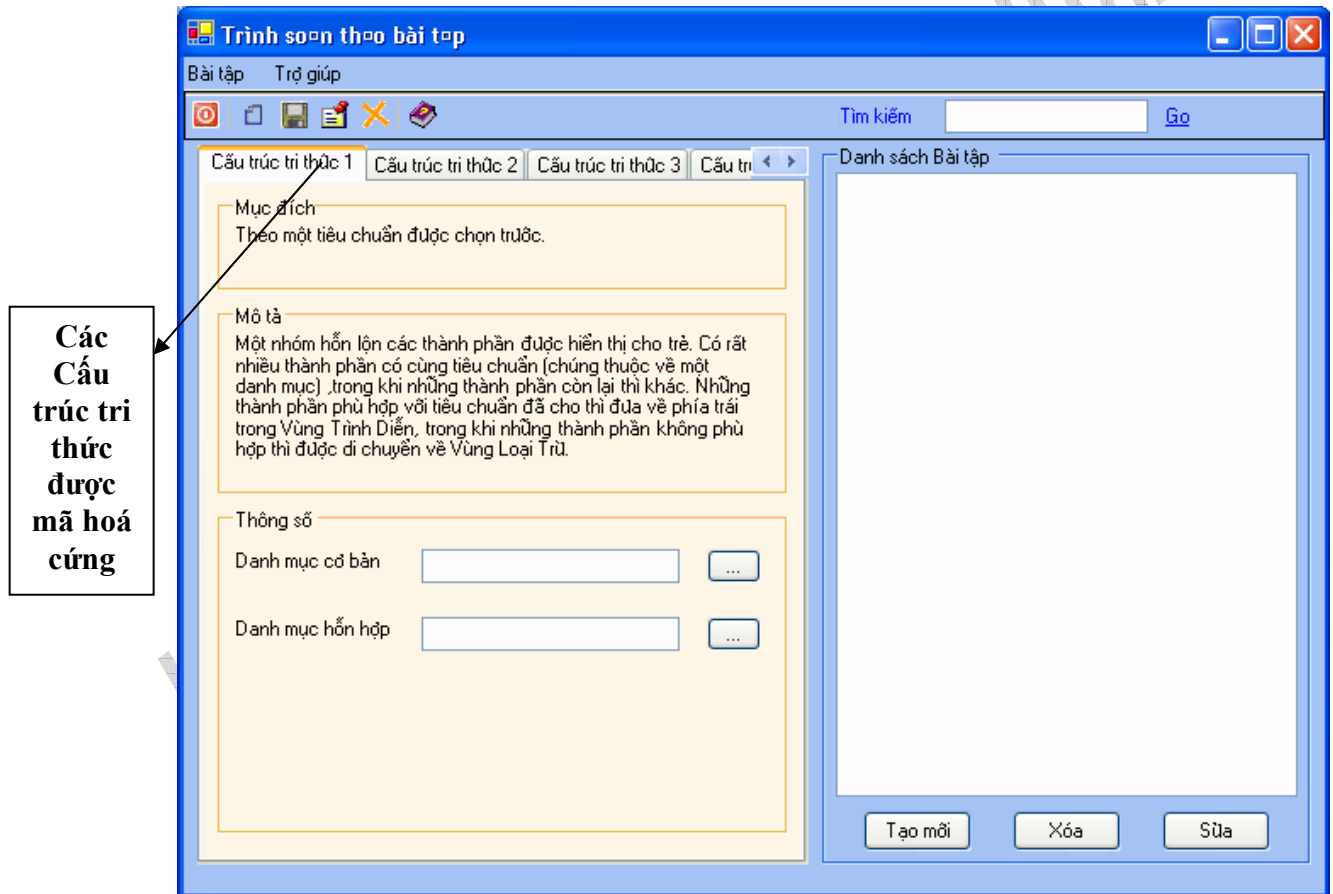


H3.9: Kiến trúc hệ thống và  
Sự tương tác giữa người sử dụng và hệ thống

### 3.7.1/ Cấu trúc tri thức ( Tri thức lĩnh vực ) :

Từ hệ thống được xây dựng mà có các cách khác nhau để mã hoá tri thức trong hệ thống ITS. Tri thức có thể được mã hoá cứng (tri thức nhúng – không thể sửa đổi - phụ thuộc lĩnh vực) hoặc mã hoá mềm (dùng các kỹ thuật để biểu diễn tri thức - độc lập lĩnh vực) v.v... Nhưng cách mã hoá tri thức đầu tiên tương đối đơn giản, ít tốn kém để xây dựng nhưng lại không thể sửa đổi hay dùng cho các lĩnh vực khác. Còn cách sau tương đối phức tạp hơn, khó xây dựng hơn nhưng lại có lợi ích lớn là có thể tái sử dụng để áp dụng cho nhiều lĩnh vực khác nhau.

Trong hệ thống mà khóa luận này xây dựng, tri thức được mã hoá dưới dạng “vừa cứng vừa mềm” và được cài đặt thành từng lớp đối tượng riêng biệt không phụ thuộc lẫn nhau; và mỗi cấu trúc tri thức được mã hoá một phần “cứng” trên giao diện người dùng (trình soạn thảo) (Hình 3.10) và một phần “mềm” để xử lý những hoạt động liên quan đến chúng. Một thuận lợi của cách này là dễ cài đặt, chỉnh sửa, cũng như thêm vào một tri thức mới (cùng lĩnh vực).

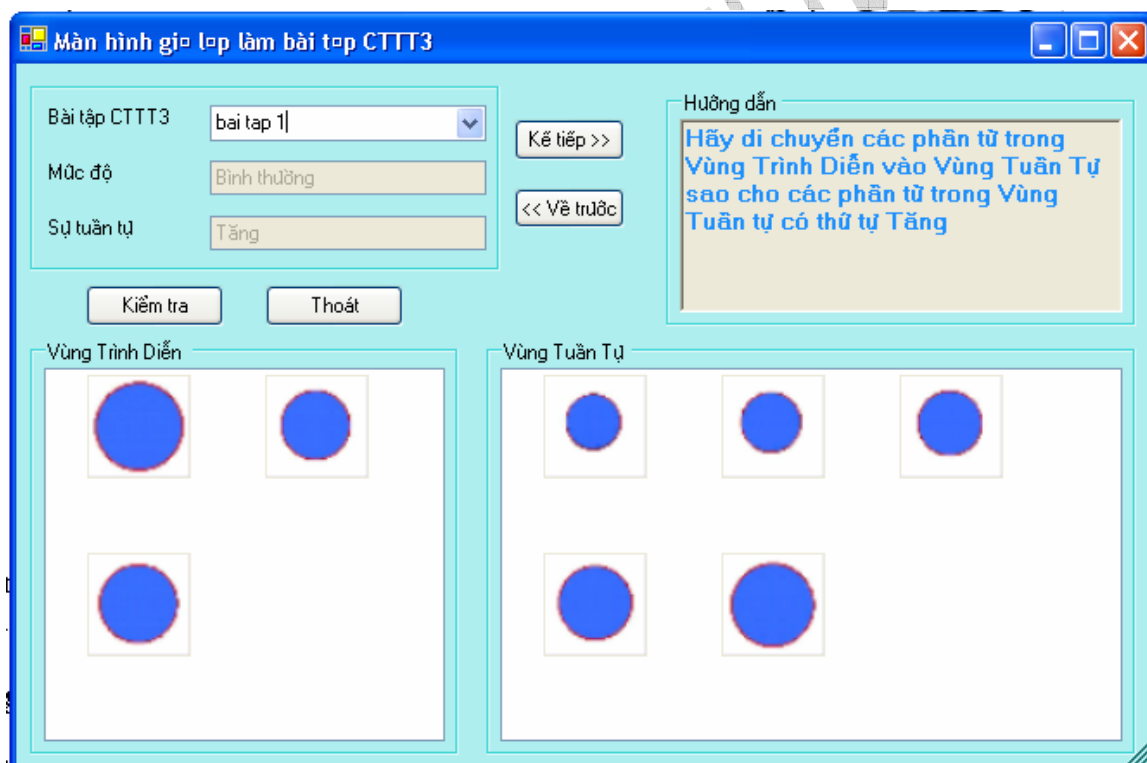


### H3.10 Cấu trúc tri thức gắn với giao diện soạn thảo

#### 3.7.2/ Bộ giả lập :

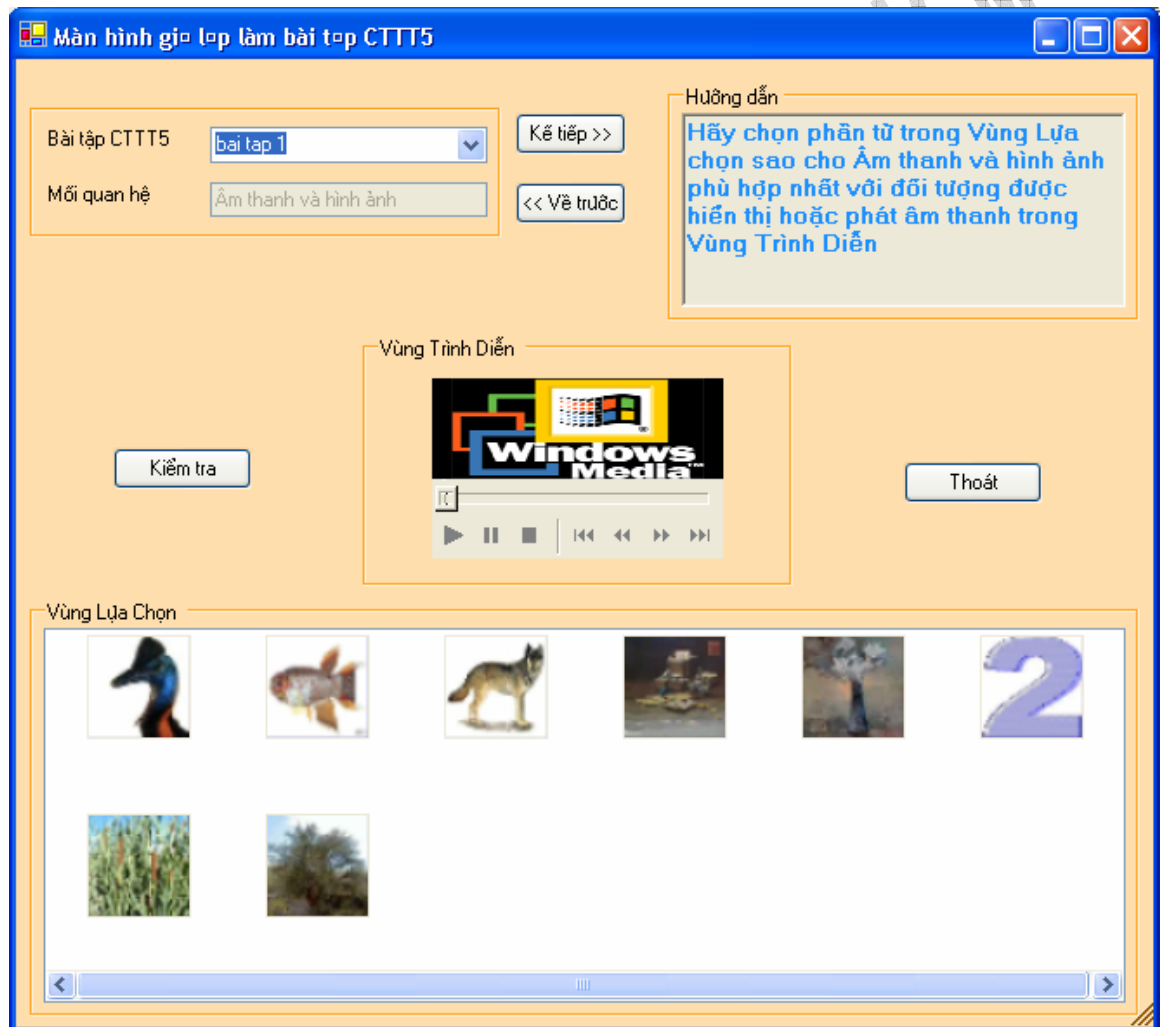
Hoạt động của bộ giả lập là thiết lập một giao diện làm bài tập cho trẻ. Giao diện này phải thân thiện, xác với ngữ cảnh thực tế mà trẻ hay gặp. Để có thể hấp dẫn trẻ trong quá trình học thì giao diện nên kết hợp nhiều hình thức trình bày khác nhau như : dùng văn bản (text) để hướng dẫn, hoặc hình ảnh hoá các bài tập sao cho dễ hiểu và thu hút trẻ, hoặc dùng âm thanh xen vào để cho quá trình học luôn sôi động, không nhàm chán ... Chẳng hạn, sau đây là một vài giao diện mà hệ thống trong khóa luận

này đã xây dựng : giao diện bài tập sử dụng Cấu trúc tri thức về sự tuần tự (Hình 3.11) – dùng hình ảnh để chỉ sự tuần tự, hoặc giao diện bài tập sử dụng Cấu trúc tri thức về



sự liên kết (Hình 3.12) - sử dụng âm thanh kết hợp hình ảnh.

### H3.11 Giao diện giả lập bài tập sử dụng Cấu trúc tri thức về sự tuần tự



### H3.12 Giao diện giả lập bài tập sử dụng Cấu trúc tri thức về sự liên kết

Tất cả quá trình làm bài tập của trẻ sẽ được kiểm soát bởi Module theo dõi và hướng dẫn. Module này kết hợp với hồ sơ của trẻ đã được lưu trữ trước đó trong Bộ đánh giá để điều chỉnh độ phức tạp của bài tập cho trẻ tùy theo từng cấp độ khác nhau của trẻ. Module cũng định rõ nội dung bài tập sẽ được sắp theo thứ tự như thế nào, loại phản hồi nào được đưa ra, khi nào và như thế nào để dạy, giải thích, sửa sai, đưa ra



hướng giải quyết ... Đây là một trong những module quan trọng góp phần làm cho hệ thống trở nên thông minh. Bởi vì ITS là những hệ thống hướng dẫn thông minh nên nó không trực tiếp đưa ra lời giải, mà chỉ cung cấp những hướng dẫn cần thiết để người học có thể tìm ra lời giải cho vấn đề của họ. Do đó Module theo dõi và hướng dẫn là một module không thể thiếu cho một ITS hoàn chỉnh. Một Module theo dõi và hướng dẫn hiệu quả là một Module mà có thể đưa ra những lời hướng dẫn hay giải thích ngay lúc mà người học gặp rắc rối trong quá trình rèn luyện của mình.

Trong ứng dụng mà khóa luận này xây dựng, các chiến lược hướng dẫn được “gắn” trực tiếp vào từng Cấu trúc tri thức khác nhau của lĩnh vực. Vì vậy mỗi một Cấu trúc tri thức (tương ứng với dạng bài tập thuộc loại này) sẽ có những chiến lược hướng dẫn khác nhau. Chẳng hạn, Cấu trúc tri thức về sự tuần tự sẽ có chiến lược hướng dẫn về sự tuần tự, hay Cấu trúc tri thức về sự liên kết sẽ có chiến lược hướng dẫn về sự liên kết. Do đó các chiến lược hướng dẫn sẽ độc lập nhau, làm cho chúng có thể được mã hoá dễ dàng hơn, nhưng có thể tốn kém do không tận dụng được sự hướng dẫn của nhau.

### **3.7.3/ Bộ soạn thảo :**

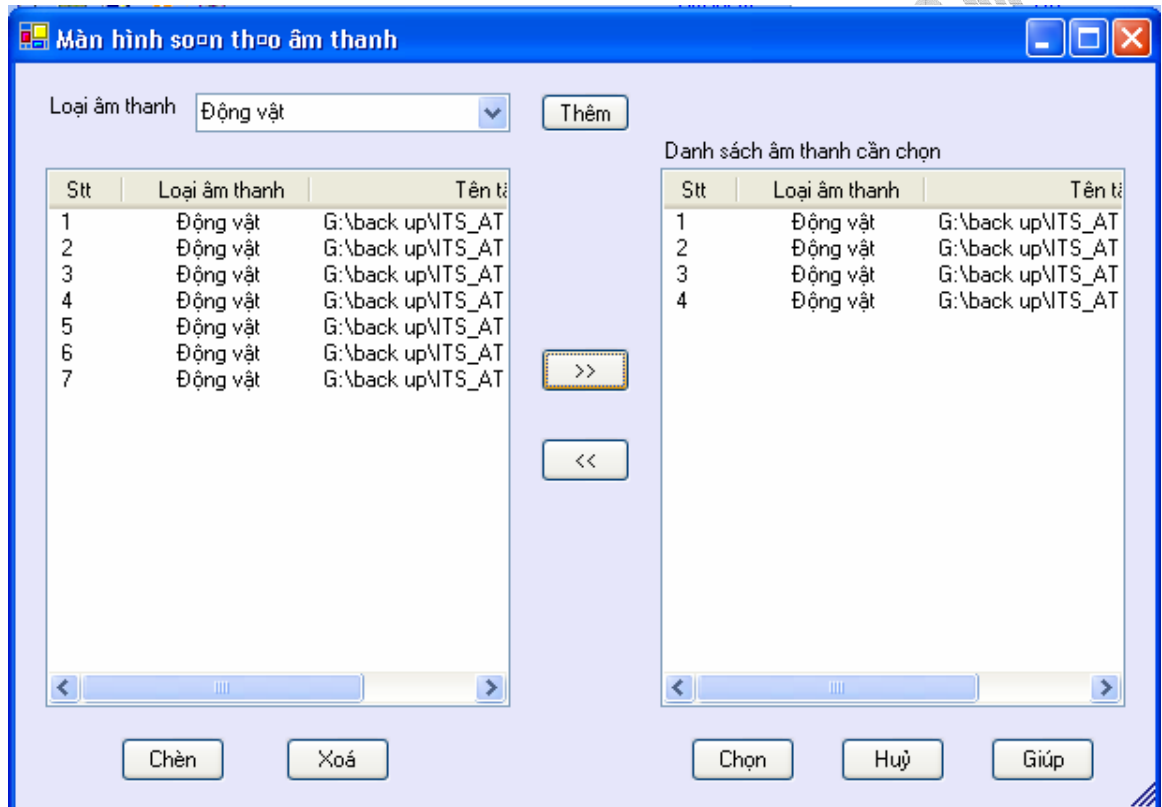
Bộ soạn thảo cho phép giáo viên tạo ra những bài tập rèn luyện cho trẻ từ những tri thức đã được mã hoá trong hệ thống. Tùy theo nội dung lĩnh vực của bài tập cần soạn thảo, chẳng hạn: sự nhận biết về màu sắc, sự nhận biết con chữ..., giáo viên có thể soạn thảo vô số bài tập khác nhau mà mình muốn trình bày cho trẻ. Tùy từng trình độ về sự nhận thức của lứa tuổi này mà giáo viên phải soạn thảo ra những bài tập phù hợp với trẻ dựa vào chuyên môn của mình.

Giao diện soạn thảo cũng phải dễ hiểu, dễ sử dụng cho giáo viên, là những người chỉ có chuyên môn dạy học, không phải là những chuyên gia về máy tính, có thể tạo ra bài tập một cách nhanh chóng và chính xác. Tất nhiên, một vài khái niệm giáo viên cần phải biết trước khi sử dụng hệ thống. Chẳng hạn, Hình 3.13 diễn tả một màn hình soạn thảo cho việc định rõ các đối tượng có sử dụng Cấu trúc tri thức về sự tuần

tự, tức hình thành và hoàn chỉnh sự tuần tự. Hình 3.14 minh họa màn hình soạn thảo xác định các đối tượng âm thanh có sử dụng cấu trúc tri thức và sự liên kết, tức hình thành những sự liên kết chính xác ...



H3.13 Trình soạn thảo cho đối tượng với Cấu trúc tri thức tuần tự



### H3.14 Trình soạn thảo cho việc xác định các đối tượng âm thanh với Cấu trúc tri thức về sự liên kết

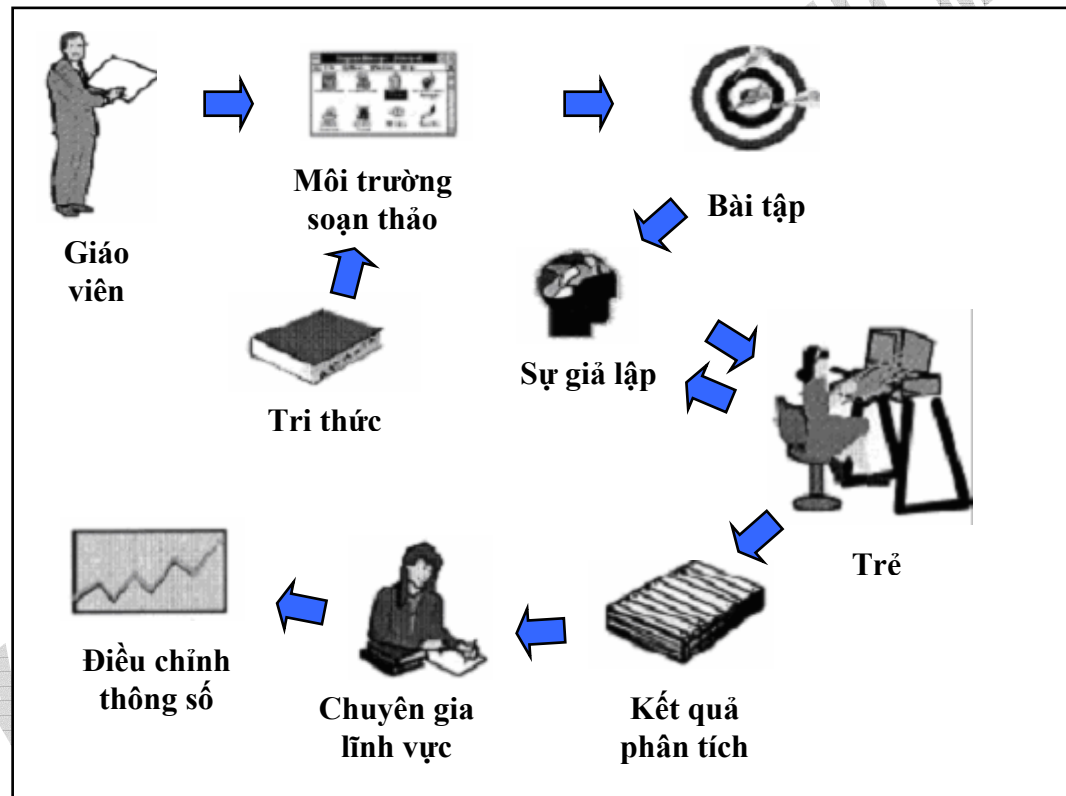
#### 3.7.4/ Bộ đánh giá :

Bộ đánh giá của hệ thống hiển thị kết quả của khả năng của trẻ (hoặc một nhóm trẻ). Những thông số thống kê đã được cập nhật thông qua sự tương tác của trẻ và hệ thống. Dữ liệu này phải được hiển thị có tính đồ họa cho giáo viên, nhà sư phạm, hoặc chuyên gia, để phân tích quá trình học của trẻ. Nó cũng cho phép sắp đặt lại những thông số chung của hệ thống, ví dụ như độ trễ thời gian được phân tích cho việc mời một trẻ tiến hành khi không có công việc nào được nhận ra trong khoảng thời gian đó và những thông số riêng biệt cho hoạt động, ví dụ như cấp độ yêu cầu, chiều dài khoảng thời gian làm bài, thiết lập chiều, .... Kể từ khi những quy luật được thông số

hóa, những thông số riêng biệt của những hoạt động cho phép sắp xếp lại những bài tập mà không cần thay đổi mã nguồn của hệ thống.

### **3.8/ Sự mô tả xử lý bên trong hệ thống :**

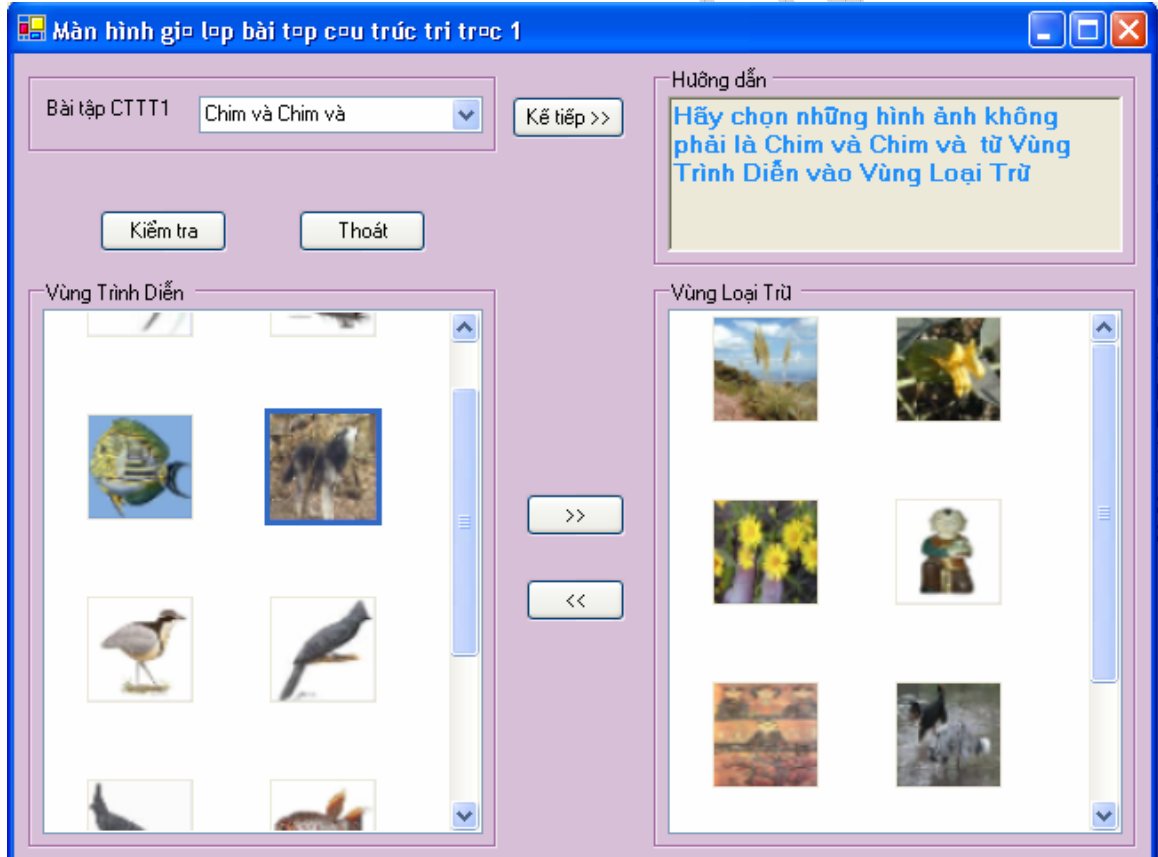
Quá trình xử lý bắt đầu với việc soạn thảo (Hình 3.15). Giáo viên có một trường soạn thảo để tạo ra bài tập. Giáo viên có thể sử dụng một trong những Cấu trúc tri thức được lưu trữ để soạn thảo nội dung bài tập giảng dạy dưới dạng các danh mục, tuần tự và lựa chọn. Ở đây, giáo viên có thể phối hợp chặt chẽ giữa hình ảnh và hình ảnh, hoặc hình ảnh và âm thanh ... để tạo ra vô tận những bài tập khác nhau trong nhiều lĩnh vực nhỏ khác nhau từ những Cấu trúc tri thức được cung cấp bởi hệ thống và chỉ bị giới hạn về không gian lưu trữ.



**H3.15: Quá trình xử lý và người dùng trong hệ thống**

Một khi những bài tập đã được tạo, trẻ có thể dùng Bộ giả lập để thực hành. Nhưng trước tiên, trẻ phải đăng nhập để hệ thống xác định khả năng của trẻ đã được Bộ đánh giá lưu trữ trước đó để đưa ra những bài tập phù hợp với trình độ của trẻ. Và quá trình làm bài tập bắt đầu khi màn hình làm bài tập dạng đầu tiên xuất hiện (trong hệ thống xây dựng là màn hình có sử dụng Cấu trúc tri thức về sự loại trừ) (Hình 3.16). Trẻ có thể chuyển đổi qua lại giữa những màn hình làm bài tập của các dạng khác nhau và chọn những bài tập mà chúng thích (hoặc theo sự hướng dẫn của giáo viên). Trong quá trình làm bài, trẻ sẽ được “quan sát và theo dõi” bởi “Module theo dõi và hướng dẫn”. Module này sẽ đưa ra những lời hướng dẫn cần thiết để trẻ có thể hoàn thành bài tập đang làm (Module này không đưa ra đáp án cuối cùng). Khi kết thúc một phiên làm

bài tập, hệ thống sẽ cho giáo viên (hay chuyên gia lĩnh vực ) biết bảng tóm lược những gì mà trẻ đã làm được.

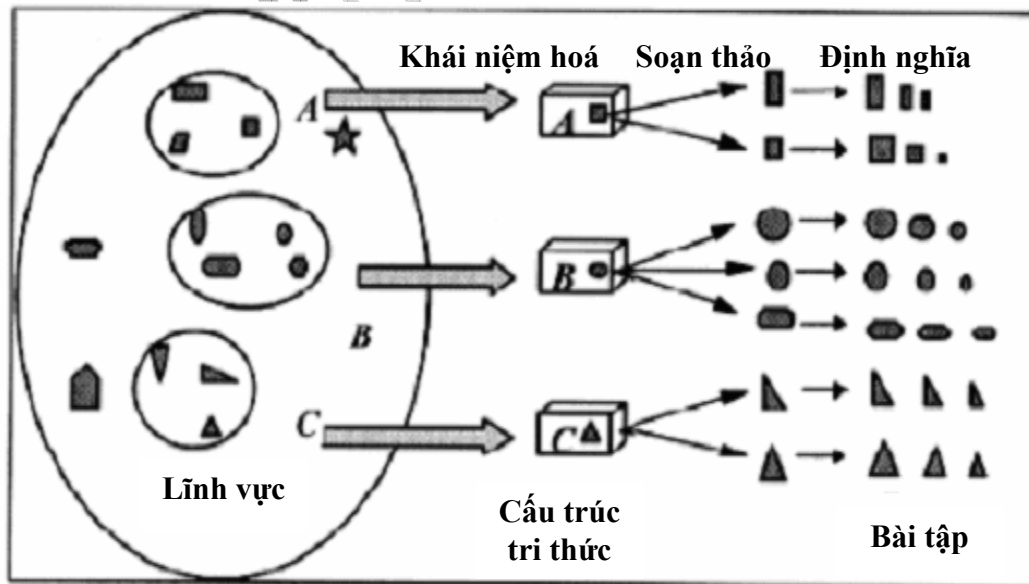


**H3.16 Màn hình giả lập bài tập Dạng đầu tiên mà Trẻ làm**

Sau phiên học, giáo viên (hay chuyên gia lĩnh vực) thông qua Bộ đánh giá để phân tích mức độ đạt được của trẻ dựa trên so sánh thông tin được lưu trữ trên Cơ sở dữ liệu. Dữ liệu này giúp cho giáo viên (hay chuyên gia lĩnh vực) có thể điều chỉnh các thông số cần thiết của hệ thống cho các phiên làm bài tập sau này.

### Kết luận và Hướng phát triển

Khóa luận giới thiệu một phương pháp cho phép định nghĩa những quy luật dùng để chuyên biệt hóa những kiến thức trong lĩnh vực được đưa, giống như một cấu trúc tri thức được định nghĩa trước. Sự hình thành đa dạng của bài tập hội tụ trong một vài cấu trúc tri thức mà nó xác định những bản chất và chức năng của bài tập. Mỗi bài tập, ít tổng quát hơn một cấu trúc tri thức, được xác định bằng sự liên kết của những cấu trúc tri thức với nội dung riêng gồm nhiều mẫu đa dạng. (Hình 3.17)



**H.3.17: Từ lĩnh vực đến bài tập**

Tại sao việc đưa ra những cấu trúc chung được thuyết minh theo nhiều cách lại hấp dẫn đến như vậy? Các nghiên cứu đã chỉ ra rằng một vấn đề càng nhiều ý nghĩa cho trẻ thì khả năng tri thức được yêu cầu và sử dụng trong ngữ cảnh khác càng cao. Nói chung thì con người sẽ học tốt hơn khi họ cảm thấy bị thu hút với đề tài được đưa, tiếp thu những khả năng cần thiết để giải quyết vấn đề. Kể từ khi có sự khác biệt tự nhiên giữa những cá thể cần thiết phải có nhiều loại bài tập khác nhau đủ để hấp dẫn tất

cả mọi người. Thêm vào đó, hệ thống giảng dạy yêu cầu một phạm vi rộng của chiến lược giảng dạy được lưu trữ trong thư viện. Những cái này nên hỗ trợ những bài tập mang tính tương tượng cho tất cả các loại trẻ để mỗi người có thể đạt tới khả năng tối đa của mình.

Chức năng của hệ thống giảng dạy không nên được giới hạn là chỉ huấn luyện. Hệ thống có thể theo dõi và đánh giá khả năng của từng trẻ và chia sẻ thông tin này cho giáo viên cũng như chuyên gia lĩnh vực. Điều này cho phép giáo viên hoặc chuyên gia lĩnh vực có thể đánh giá lại sự kiên định của hệ thống và sự hoàn chỉnh để cải tiến lại cơ sở tri thức và những thông số của bài tập.

Đây là một đề tài lý thú cả trong việc nghiên cứu cũng như ứng dụng. Nếu hệ thống được hoàn chỉnh thì nó sẽ mang hiệu quả rất lớn trong việc giảng dạy. Tuy nhiên, những hệ thống này đòi hỏi phải bỏ nhiều công sức của nhiều người trong nhiều lĩnh vực khác nhau như nhà sư phạm, nhà tâm lý học sư phạm, chuyên gia lập trình ... cũng như nhiều thời gian và chi phí để xây dựng nên một hệ thống hoàn chỉnh.

Khóa luận này đã đưa ra một kiến trúc chung cho toàn bộ hệ thống cần xây dựng. Nhưng khóa luận chỉ mới xây dựng được một phần của hệ thống. Đó là Bộ soạn thảo - đây là Module dùng để hỗ trợ cho giáo viên soạn thảo bài tập cho môn học của mình dựa vào những Cấu trúc tri thức đã được mã hoá và lưu trữ trong hệ thống. Mỗi bài tập sẽ tương ứng với một Cấu trúc tri thức xác định bản chất và tính năng của bài tập. Trẻ sẽ làm các bài tập này trên Bộ giả lập mà hệ thống đã xây dựng. Tuy nhiên, Bộ giả lập chưa thật hoàn chỉnh ở Module hướng dẫn. Khóa luận này chỉ mới xây dựng được một phần Module này. Nhưng với những gì đã xây dựng, hệ thống đã thực hiện được những chức năng cơ bản của hệ thống ITS. Tất cả kết quả của trẻ sẽ được đánh giá và nhận xét bởi Bộ đánh giá mà hệ thống sẽ xây dựng tiếp theo. Nhờ vào Module này mà giáo viên có thể theo dõi quá trình học của trẻ để có thể cập nhật kịp thời.



Vi vậy, để hệ thống hoàn thiện hơn và có thể áp dụng trong thực tế thì cần phải nhiều nỗ lực hơn nữa. Trước tiên là, nghiên cứu kỹ hơn về lĩnh vực đã chọn cho phù hợp với tâm lý và khả năng của trẻ ở nước ta. Thứ hai là cần hoàn thiện thêm đầy đủ Cấu trúc tri thức trong Mô hình lĩnh vực. Và điều quan trọng nhất của hệ thống là nghiên cứu và xây dựng Mô hình hướng dẫn và Mô hình đánh giá phù hợp. Đây là một công việc khó khăn, thích hợp cho những ai nghiên cứu về Tri thức, Trí tuệ nhân tạo. Rất hi vọng các khoá sau có thể tiếp tục công việc và xây dựng một hệ thống hoàn chỉnh có thể áp dụng thực tế. Vì như đã nói nếu hệ thống được hoàn thành thì đây là một tiền đề tốt để có thể nghiên cứu và xây dựng những hệ thống trong nhiều lĩnh vực khác, chẳng hạn : Hệ thống ITS giảng dạy ngôn ngữ lập trình có thể áp dụng cho các sinh viên mới học lập trình...

**Tài liệu tham khảo :**

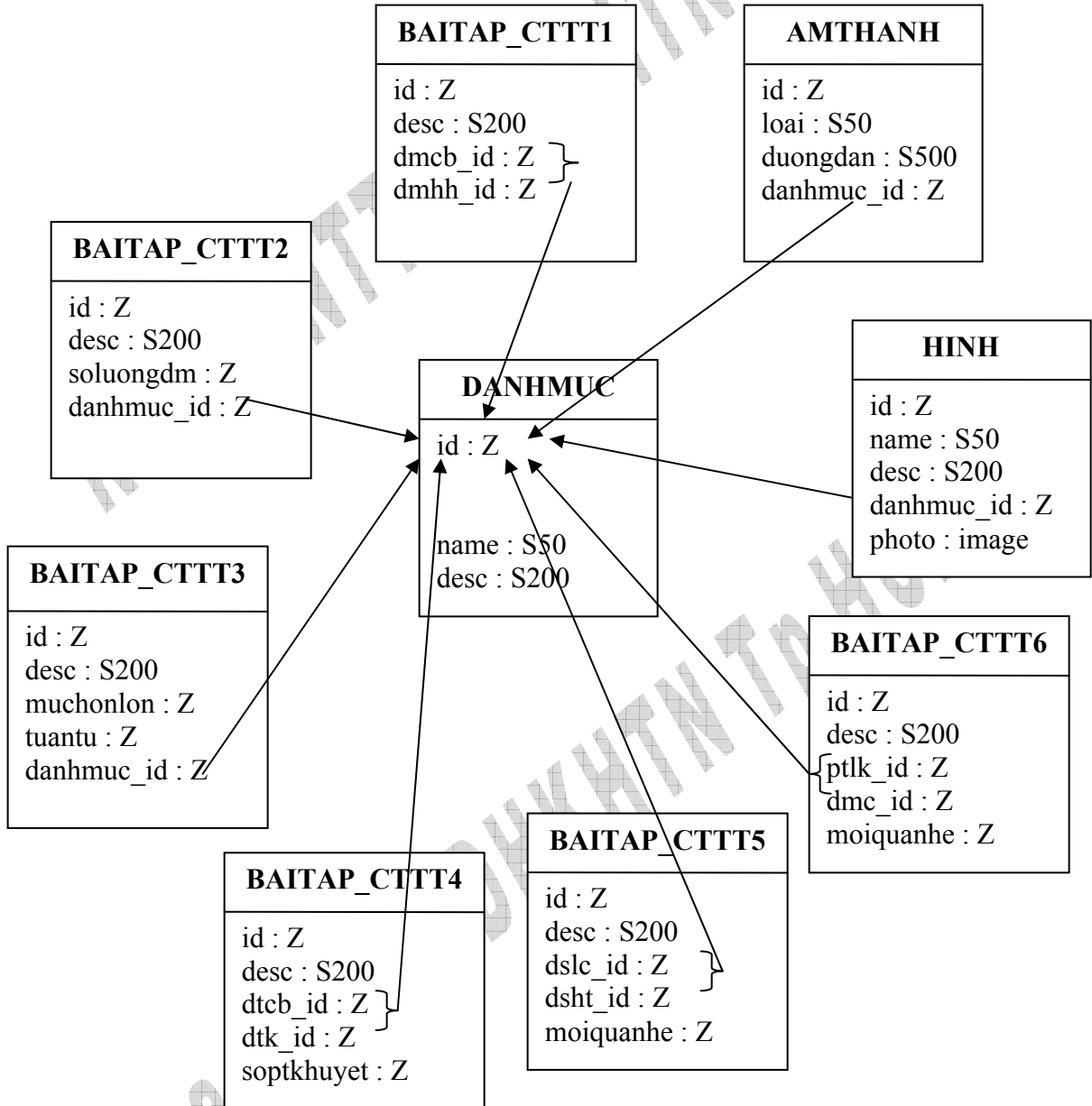
- [1] Chiu-Chen Hsieh, Tzong-Han Tsai, David Wible, Wen-Lian Hsu, Exploiting Knowledge Representation in an Intelligent Tutoring System for English Lexical Errors, Proceedings of the International Conference on Computers in Education (ICCE'02), 2002.
- [2] Mark Urban-Lurain, An Historic Review in the Context of the Development of Artificial Intelligence and Education Psychology, Michigan State University, [www.cse.msu.edu/rgroups/cse101/ITS/its.htm](http://www.cse.msu.edu/rgroups/cse101/ITS/its.htm)
- [3] Miguel Nussbaum, Ricardo Rosas, Isabel Peirano, Francisco Cardenas, Development Intelligent Tutoring Systems using Knowledge Structures, Computers & Education (2001), 36, p15-32.
- [4] Randall Spark, Scott Dooley, Lori Meiskey, Rick Blumenthal , The LEAP Authoring Tool : Supporting complex courseware authoring through reuse, rapid prototyping, and interactive visualizations, International Journal of Artificial Intelligence in Education (1998), 10, p75-97.
- [5] Shamus Paul Smith, Developing An Authoring Environment For Procedural Task Tutoring System, PhD Thesis. Massey University, New Zealand, 1997,3.
- [6] Steven Linton, Knowledge Representation and Acquisition for Intelligent Tutoring System, Computer Science & Software Engineering, University of Canterbury, [www.cose.canterbury.ac.nz/research/reports/HonsReps/2003](http://www.cose.canterbury.ac.nz/research/reports/HonsReps/2003)
- [7] Tom Murray, Authoring Intelligent Tutoring Systems: An analysis of State of Art, International Journal of Artificial Intelligence in Education (1999), 10, p98-129.
- [8] Yujian Zhou, Martha W.Evens, A Practical Student Model in an Intelligent Tutoring System, Proceedings of 11<sup>th</sup> IEEE International Conference on Tools with Artificial Intelligence, Chicago, IL, 1999, p13-18.

**Phụ lục :**

*Hồ sơ thiết kế phần mềm*

**1/ Thiết kế dữ liệu :**

**1.1/ Sơ đồ logic dữ liệu :**



GIAOVIEN	HOCSINH	CHUYEN GIA
MaGiaoVien: S50 TenGiaoVien: S50 NgaySinh: Date DiaChi: S50 TenDangNhap:S50 MatKhau: S50	MaHocSinh: S50 TenHocSinh: S50 NgaySinh: Date DiaChi: S50 TenDangNhap:S50 MatKhau: S50 TrinhDo: N	MaChuyenGia:S50 TenChuyenGia:S50 NgaySinh: Date DiaChi: S50 TenDangNhap:S50 MatKhau: S50

### 1.2/ Thiết kế dữ liệu :

a/ DANHMUC : id, name, desc

STT	Tên cột	Kiểu	Ràng buộc	Ghi chú
1	id	int(4)	Khoá chính	
2	name	varchar(50)		
3	desc	varchar(200)		

b/ HINH : id, name, desc, danhmuc\_id, photo

STT	Tên cột	Kiểu	Ràng buộc	Ghi chú
1	id	int(4)	Khoá chính	
2	name	varchar(50)		
3	desc	varchar(200)		
4	danhmuc id	Int(4)	Khoá ngoại	
5	photo	Image		

c/ AMTHANH : id, name, desc, danhmuc\_id, photo

STT	Tên cột	Kiểu	Ràng buộc	Ghi chú
1	id	int(4)	Khoá chính	
2	name	varchar(50)		
3	desc	varchar(200)		
4	danhmuc id	Int(4)	Khoá ngoại	
5	photo	Image		

d/ BAITAP\_CTTT1 : id, desc, dmcb\_id, dmhh\_id

STT	Tên cột	Kiểu	Ràng buộc	Ghi chú
-----	---------	------	-----------	---------

1	Id	int(4)	Khoá chính	
2	desc	Varchar(200)		
3	dmcb_id	Int(4)	Khoá ngoại	
4	dmhh_id	Int(4)	Khoá ngoại	

e/ **BAITAP\_CTTT2** : id, desc, soluongdm, danhmuc\_id

STT	Tên cột	Kiểu	Ràng buộc	Ghi chú
1	Id	int(4)	Khoá chính	
2	desc	Varchar(200)		
3	Soluongdm	Int(4)		
4	Danhmuc_id	Int(4)	Khoá ngoại	

f/ **BAITAP\_CTTT3** : id, desc, muchonlon,tuantu, danhmuc\_id

STT	Tên cột	Kiểu	Ràng buộc	Ghi chú
1	Id	int(4)	Khoá chính	
2	desc	Varchar(200)		
3	Muchonlon	Int(1)		
4	Tuantu	Int(1)		
5	Danhmuc_id	Int(4)	Khoá ngoại	

g/ **BAITAP\_CTTT4** : id, desc, dtcb\_id, dtk\_id, soptkhuyet

STT	Tên cột	Kiểu	Ràng buộc	Ghi chú
1	Id	int(4)	Khoá chính	
2	desc	Varchar(200)		
3	Dtcb_id	Int(4)	Khoá ngoại	
4	Dtk_id	Int(4)	Khoá ngoại	
5	Soptkhuyet	Int(4)		

h/ **BAITAP\_CTTT5** : id, desc, dslc\_id, dsht\_id, moiquanhe

STT	Tên cột	Kiểu	Ràng buộc	Ghi chú
1	Id	int(4)	Khoá chính	
2	Desc	Varchar(200)		
3	Dslc_id	Int(4)	Khoá ngoại	
4	Dsht_id	Int(4)	Khoá ngoại	
5	Moiquanhe	Int(1)		

k/ **BAITAP\_CTTT6** : id, desc, ptlk\_id, dmc\_id, moiquanhe

STT	Tên cột	Kiểu	Ràng buộc	Ghi chú
1	id	int(4)	Khoá chính	
2	desc	Varchar(200)		
3	ptlk_id	Int(4)	Khoá ngoại	
4	dmc_id	Int(4)	Khoá ngoại	
5	moiquanhe	Int(1)		

l/ **GIAOVIEN**: MaGiaoVien, TenGiaoVien, NgaySinh, DiaChi, TenDangNhap, MatKhau.

STT	Tên cột	Kiểu	Ràng buộc	Ghi chú
1	MaGiaoVien	varchar(50)	Khoá chính	
2	TenGiaoVien	varchar(50)		
3	NgaySinh	Date		
4	DiaChi	Varchar(50)		
5	TenDangNhap	Varchar(50)		
6	MatKhau	Varchar(50)		

m/ **HOCSINH**: MaHocSinh, TenHocSinh, NgaySinh, DiaChi, TenDangNhap, MatKhau, TrinhDo.

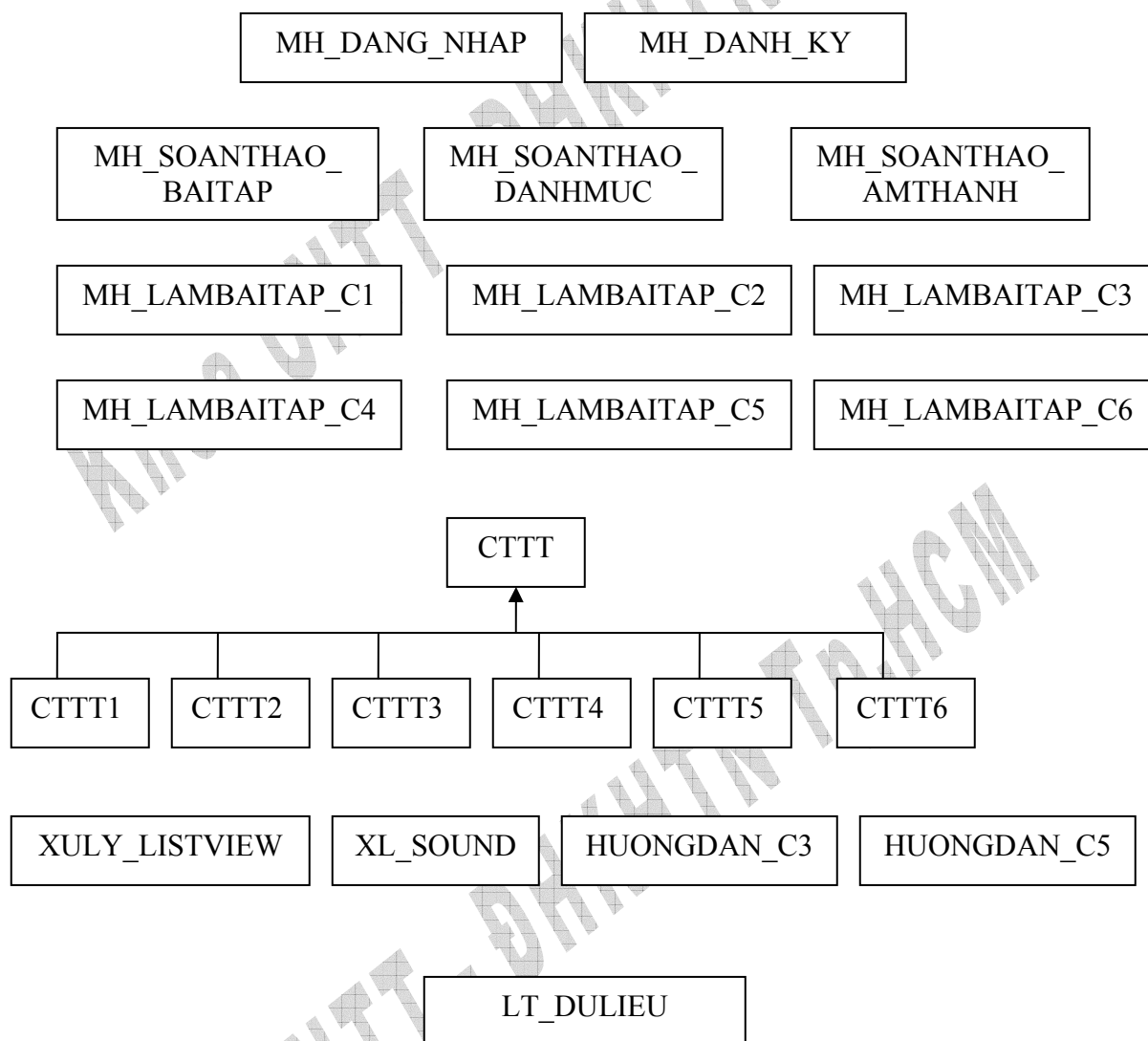
STT	Tên cột	Kiểu	Ràng buộc	Ghi chú
1	MaHocSinh	varchar(50)	Khoá chính	
2	TenHocSinh	varchar(50)		
3	NgaySinh	Date		
4	DiaChi	Varchar(50)		
5	TenDangNhap	Varchar(50)		
6	MatKhau	Varchar(50)		
7	TrinhDo	Int		

n/ **CHUYENGIA**: MaChuyenGia, TenChuyenGia, NgaySinh, DiaChi, TenDangNhap, MatKhau.

STT	Tên cột	Kiểu	Ràng buộc	Ghi chú
1	MaChuyenGia	varchar(50)	Khoá chính	
2	TenChuyenGia	varchar(50)		
3	NgaySinh	Date		
4	DiaChi	Varchar(50)		

5	TenDangNhap	Varchar(50)		
6	MatKhau	Varchar(50)		

**2/ Mô hình kiến trúc tổng thể :**



**3/ Thiết kế xử lý :****3.1/ Mô tả chi tiết lớp lưu trữ dữ liệu XL\_DULIEU :****Danh sách các biến thành phần**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	Sqlconn	SqlConnection	biến kết nối vào CSDL	
2	Sqlcmd	SqlCommand	biến thực hiện lệnh truy vấn	
3	strConn	String	chuỗi kết nối	
4	nImage	Int	số lượng hình	
5	nDanhMuc		số lượng danh mục	

**Danh sách các hàm thành phần**

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	LT_DuLieu()			nImage=0; nDanhMuc=0;	Hàm khởi tạo
2	MoKetNoi()			strConn="Integrated Security=SSPI;Initial Catalog=ITSDB;Data Source=localhost;Workstation ID=localhost;"; Sqlconn = new SqlConnection(strConn); Sqlconn.Open();	Mở kết nối
3	DongKetNoi()			Sqlconn.Close();	Đóng kết nối
4	IsConnOpen()		Bool	if(Sqlconn.State == ConnectionState.Open) return true; return false;	kiểm tra mở
5	IsConnClosed()		bool	if(Sqlconn.State == ConnectionState.Closed) return true; return false;	kiểm tra đóng
6	ThemGV()	string MaGV,string TenGV,string Ns,string Dc,string Tendn,string Mk		Sqlcmd.Parameters.Add( "@MaGV",SqlDbType.VarChar,50) .Value = MaGV; Tương tự cho các tham số còn lại	Thêm giáo viên vào bảng GIAOVIEN
7	ThemCG	string MaCG,string		Sqlcmd.Parameters.Add( "@MaCG",SqlDbType.VarChar,50)	Thêm chuyên gia vào bảng



		TenCG,string Ns,string Dc,string Tendn,string Mk		.Value = MaCG; Tương tự cho các tham số còn lại	CHUYENGIA
8	ThemHS()	string MaHS,string TenHS,string Ns,string Dc,string Tendn,string Mk,int td		SqlCommand.Parameters.Add( "@MaCG",SqlDbType.VarChar,50) .Value = MaCG; Tương tự cho các tham số còn lại	Thêm học sinh vào bảng HOCSINH
9	LayTenHocSinh()		SQLDataReader	SqlCommand = new SqlCommand( "Select * From HOCSINH", Sqlconn) Return SqlCommand.ExecuteReader()	Lấy ra tất cả tên học sinh trong bảng HOCSINH
10	LayTenGiaoVien()		SQLDataReader	SqlCommand = new SqlCommand( "Select * From GIAOVIEN", Sqlconn) Return SqlCommand.ExecuteReader()	Lấy ra tất cả tên giáo viên trong bảng GIAOVIEN
11	LayTenChuyenGia()		SQLDataReader	SqlCommand = new SqlCommand( "Select * From CHUYENGIA", Sqlconn) Return SqlCommand.ExecuteReader()	Lấy ra tất cả tên chuyên gia trong bảng CHUYENGIA
12	KiemTraUserName()	String username, SqlDataReader dr		while(dr.Read()) .... Return co return khongco	Kiểm tra Tên đăng nhập đó có tồn tại hay không
13	LaySL()	String StrSQL	Int		đếm số lượng thành viên trong mỗi loại đăng nhập
14	PhatSinhMa	String KTDau, string StrSQL)	String	Ma=KTDau+sl.toString()	Phát sinh mã tự động
15	InsertBaitap_C1()	int id,string desc, int dmcb, int dmhh		SqlCommand = new SqlCommand("sp_InsertBaitap_C1", Sqlconn); SqlCommand.CommandType = CommandType.StoredProcedure; ... SqlCommand.ExecuteNonQuery();	Thêm một bài tập vào bảng BAITAP_ CTTT1
16	LaySoLuongBT_CT1()		int	SqlCommand = new SqlCommand("Select count(id) From BAITAP_CTTT1",Sqlconn); Object obj=SqlCommand.ExecuteScalar(); return	Lấy số lượng trong bảng BAITAP_ CTTT1

				<code>int.Parse(obj.ToString());</code>	
17	LayBaiTapCTTT1()		Sql Data Reader	<code>SqlCommand = new SqlCommand("Select * From BAITAP_CTTT1", SqlConnection); return SqlCommand.ExecuteReader();</code>	Lấy tất cả bài tập trong bảng BAITAP_CTTT1
18	LayDMCB1()	int baitap_id	Sql Data Reader	<code>SqlCommand= new SqlCommand("sp_LayDMCB1", SqlConnection); ... return SqlCommand.ExecuteReader();</code>	Lấy danh mục cơ bản trong BAITAP_CTTT1
19	LayDMHH1()	int baitap_id	Sql Data Reader	<code>SqlCommand= new SqlCommand("sp_LayDMHH1", SqlConnection); ... return SqlCommand.ExecuteReader();</code>	Lấy danh mục hỗn hợp trong BAITAP_CTTT1
20	InsertBaitap_C2()	int id,string desc, int soluong, int danhmuc_id		<code>SqlCommand = new SqlCommand("sp_InsertBaitap_C2", SqlConnection); ... SqlCommand.ExecuteNonQuery();</code>	Thêm một bài tập vào bảng BAITAP_CTTT2
21	LaySoLuongBT_CT2()		int	<code>SqlCommand = new SqlCommand("Select count(id) From BAITAP_CTTT2",SqlConnection); Object obj=SqlCommand.ExecuteScalar(); return int.Parse(obj.ToString());</code>	Lấy số lượng bài tập trong bảng BAITAP_CTTT2
22	LayBaiTapCTTT2()		Sql Data Reader	<code>SqlCommand = new SqlCommand("Select * From BAITAP_CTTT2", SqlConnection); return SqlCommand.ExecuteReader();</code>	Lấy tất cả bài tập trong bảng BAITAP_CTTT2
23	LayDanhMucC2()	int baitap_id	Sql Data Reader	<code>SqlCommand= new SqlCommand("sp_LayDanhMuc2", SqlConnection); ... return SqlCommand.ExecuteReader();</code>	Lấy danh mục trong BAITAP_CTTT2
24	LaySoLuongDM_C2()	int baitap_id	int	<code>... string s="Select soluongdm from BAITAP_CTTT2 Where id="+baitap_id.ToString(); SqlCommand = new SqlCommand(s,SqlConnection); SqlDataReader dr = SqlCommand.ExecuteReader(); ...</code>	Lấy số lượng danh mục của một bài tập dạng BAITAP_CTTT2
25	InsertBaitap_C3()	int id, string desc,byte muchonlon,byte		<code>SqlCommand = new SqlCommand("sp_InsertBaitap_C3", SqlConnection);</code>	Thêm một bài tập vào bảng BAITAP_CTTT2

		tuantu,int danhmuc_id		... Sqlcmd.ExecuteNonQuery();	CTTT3
26	LaySoLuongBT_CT3()		int	Sqlcmd = new SqlCommand("Select count(id) From BAITAP_CTTT3",Sqlconn); Object obj=Sqlcmd.ExecuteScalar(); return int.Parse(obj.ToString());	Lấy số lượng bài tập trong bảng BAITAP_ CTTT3
27	LayBaiTapCTTT3()		Sql Data Reader	Sqlcmd = new SqlCommand("Select * From BAITAP_CTTT3", Sqlconn); return Sqlcmd.ExecuteReader();	Lấy tất cả bài tập trong bảng BAITAP_ CTTT3
28	LayDanhMuc_C3()	int baitap_id	Sql Data Reader	Sqlcmd= new SqlCommand("sp_LayDanhMuc_C3", Sqlconn); ... return Sqlcmd.ExecuteReader();	Lấy danh mục trong BAITAP_ CTTT3
29	LayThongSo_C3()	int baitap_id	Sql Data Reader	string s="Select muchonlon,tuantu From BAITAP_CTTT3 Where id="+baitap_id.ToString(); Sqlcmd= new SqlCommand(s, Sqlconn); return Sqlcmd.ExecuteReader();	Lấy tất cả thông số của bài tập dạng BAITAP_ CTTT3
30	InsertBaitap_C4()	int id,string desc,int dtkb_id,int dtk_id,int soptk		Sqlcmd = new SqlCommand("sp_InsertBaitap_C4", Sqlconn); ... Sqlcmd.ExecuteNonQuery();	Thêm một bài tập vào bảng BAITAP_ CTTT4
31	LaySoLuongBT_CT4()		int	Sqlcmd = new SqlCommand("Select count(id) From BAITAP_CTTT4",Sqlconn); Object obj=Sqlcmd.ExecuteScalar(); return int.Parse(obj.ToString());	Lấy số lượng bài tập trong bảng BAITAP_ CTTT4
32	LayBaiTapCTTT4()		Sql Data Reader	Sqlcmd = new SqlCommand("Select * From BAITAP_CTTT4", Sqlconn); return Sqlcmd.ExecuteReader();	Lấy tất cả bài tập trong bảng BAITAP_ CTTT4
33	LayDanhMuc1_C4()	int baitap_id	Sql Data Reader	Sqlcmd= new SqlCommand("sp_LayDanhMuc1_C4", Sqlconn); ... return Sqlcmd.ExecuteReader();	Lấy danh mục trong BAITAP_ CTTT4

34	LayDanhMuc2_C4()	int baitap_id	Sql Data Reader	Sqlcmd= new SqlCommand("sp_LayDanhMuc2_C4", Sqlconn); ... return Sqlcmd.ExecuteReader();	Lấy danh mục trong BAITAP_ CTTT4
35	LayThongSo_C4()	int baitap_id	Sql Data Reader	string s="Select soptkhuyet From BAITAP_CTTT4 Where id="+baitap_id.ToString(); Sqlcmd= new SqlCommand(s, Sqlconn);  return Sqlcmd.ExecuteReader();	Lấy tất cả thông số của bài tập dạng BAITAP_ CTTT4
36	InsertBaitap_C5()	int id,string desc,int dslc_id,int dsht_id,int moiquanhe		Sqlcmd = new SqlCommand("sp_InsertBaitap_C5", Sqlconn); ... Sqlcmd.ExecuteNonQuery();	Thêm một bài tập vào bảng BAITAP_ CTTT5
37	LaySoLuongBT_CT5()		int	Sqlcmd = new SqlCommand("Select count(id) From BAITAP_CTTT5",Sqlconn); Object obj=Sqlcmd.ExecuteScalar(); return int.Parse(obj.ToString());	Lấy số lượng bài tập trong bảng BAITAP_ CTTT5
38	LayThongSo_C5()	int baitap_id	Sql Data Reader	string s="Select soptkhuyet From BAITAP_CTTT5 Where id="+baitap_id.ToString(); Sqlcmd= new SqlCommand(s, Sqlconn);  return Sqlcmd.ExecuteReader();	Lấy tất cả thông số của bài tập dạng BAITAP_ CTTT5
39	LayDanhMuc1_C5()	int baitap_id	Sql Data Reader	Sqlcmd= new SqlCommand("sp_LayDanhMuc1_C5", Sqlconn); ... return Sqlcmd.ExecuteReader();	Lấy danh mục trong BAITAP_ CTTT5
40	LayDanhMuc2_C5()	int baitap_id	Sql Data Reader	Sqlcmd= new SqlCommand("sp_LayDanhMuc2_C5", Sqlconn); ... return Sqlcmd.ExecuteReader();	Lấy danh mục trong BAITAP_ CTTT5
41	LayDanhMuc3_C5()	int baitap_id	Sql Data	Sqlcmd= new SqlCommand("sp_LayDanhMuc3_C5",	Lấy danh mục trong

			Reader	Sqlconn); ... return SqlCommand.ExecuteReader();	BAITAP_ CTTT5
42	InsertBaitap_C6()	int id,string desc,int dsle_id,int dsht_id,int moiquanhe		SqlCommand = new SqlCommand("sp_InsertBaitap_C6", Sqlconn); ... SqlCommand.ExecuteNonQuery();	Thêm một bài tập vào bảng BAITAP_ CTTT6
43	LaySoLuongBT_CT6()		int	SqlCommand = new SqlCommand("Select count(id) From BAITAP_ CTTT6",Sqlconn); Object obj=SqlCommand.ExecuteScalar(); return int.Parse(obj.ToString());	Lấy số lượng bài tập trong bảng BAITAP_ CTTT6
44	LayThongSo_C6()	int baitap_id	Sql Data Reader	string s="Select soptkhuyet From BAITAP_ CTTT6 Where id="+baitap_id.ToString(); SqlCommand= new SqlCommand(s, Sqlconn);  return SqlCommand.ExecuteReader();	Lấy tất cả thông số của bài tập dạng BAITAP_ CTTT6
45	LayDanhMuc1_C6()	int baitap_id	Sql Data Reader	SqlCommand= new SqlCommand("sp_LayDanhMuc1_C6", Sqlconn); ... return SqlCommand.ExecuteReader();	Lấy danh mục trong BAITAP_ CTTT6
46	LayDanhMuc2_C6()	int baitap_id	Sql Data Reader	SqlCommand= new SqlCommand("sp_LayDanhMuc2_C6", Sqlconn); ... return SqlCommand.ExecuteReader();	Lấy danh mục trong BAITAP_ CTTT6

### 3.2/ Mô tả chi tiết lớp XL\_LISTVIEW :

#### Danh sách các biến thành phần

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú

#### Danh sách các hàm thành phần

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	XoaPhanTu()	ListViewXPDrpDrop lv,ImageList imli		<pre>int i; if(lv.SelectedItems.Count != 0) {     i=lv.SelectedItems[0].ImageIndex;      foreach(ListViewItem item in lv.SelectedItems)     {         imli.Images.RemoveAt(i);     } } CapNhat_Listview(lv,imli);</pre>	Xoá một phần tử trong listview
2	CapNhat_Listview()	ListViewXPDrpDrop lv,ImageList il		<pre>lv.Clear(); for(int i=0;i&lt;il.Images.Count;i++) {     ListViewItem item=new ListViewItem("",i);     lv.Items.Add(item); }</pre>	Cập nhật lại listview sau khi xử lý
3	ThemPTImageList()	ImageList mliNguon ,ListViewXPDrpDrop lvNguon,ImageList imliDich		<pre>foreach (ListViewItem item in lvNguon.SelectedItems) {     imliDich.Images.Add (imliNguon.Images[item.ImageIndex]); }</pre>	Thêm vào imagelist các phần tử
4	ThemPTListView()	ListViewXPDrpDrop lvDich,ImageList imliDich		<pre>for(int i=lvDich.Items.Count; i&lt;imliDich.Images.Count;i++) {     ListViewItem nit = new ListViewItem("",i);     lvDich.Items.Add(nit);     lvDich.Update(); }</pre>	Thêm các phần tử vào listview
5	Them1PT()	ListViewXPDrpDrop lv,int nImage		<pre>ListViewItem newitem= new ListViewItem("",nImage); lv.Items.Add(newitem); lv.Update();</pre>	Thêm 1 phần tử vào listview

### 3.3/ Mô tả chi tiết lớp SOUND :

**Danh sách các biến thành phần**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú

**Danh sách các hàm thành phần**

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	Play()	string strFileName, PlaySoundFlags soundFlags			Play một file âm thanh
2	PlaySound ()	( string szSound, IntPtr hMod, PlaySoundFlags flags	bool		[DllImport("winmm.dll")]

**3.3/ Mô tả chi tiết lớp HUONGDAN\_C3 :****Danh sách các biến thành phần**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú

**Danh sách các hàm thành phần**

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	SoSanh2DoiTuong()	int Len1, int Len2	bool	if(Len1 >= Len2) return true; return false;	So sánh độ lớn 2 đối tượng
2	HuongDanNhoLon()	int DTMoi, int DTCu, byte Tuantu	string	string sHD=""; if(SoSanh2DoiTuong(DTMoi, DTCu)) { if(Tuantu == 0) sHD="Chúc mừng !! Em đã sắp xếp chính xác. Hãy tiếp tục"; else sHD="Wow!! Sai rồi ! Đối tượng các em vừa bỏ vào lớn hơn đối tượng đã có. Hãy làm lại và chọn lớn đối tượng lớn !";	Trả về hướng dẫn

				<pre> } else {     if(Tuantu==0)         sHD="Wow!! Sai rồi ! ĐỐI tượng các em vừa bỏ vào nhỏ hơn đối tượng đã có. Hãy làm lại và chọn đối tượng lớn hơn !";     else         sHD="Chúc mừng !! Em đã sắp xếp chính xác. Hãy tiếp tục"; } return sHD; </pre>	
--	--	--	--	--	--

## 3.4/ Mô tả chi tiết lớp HUONGDAN\_C5 :

**Danh sách các biến thành phần**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú

**Danh sách các hàm thành phần**

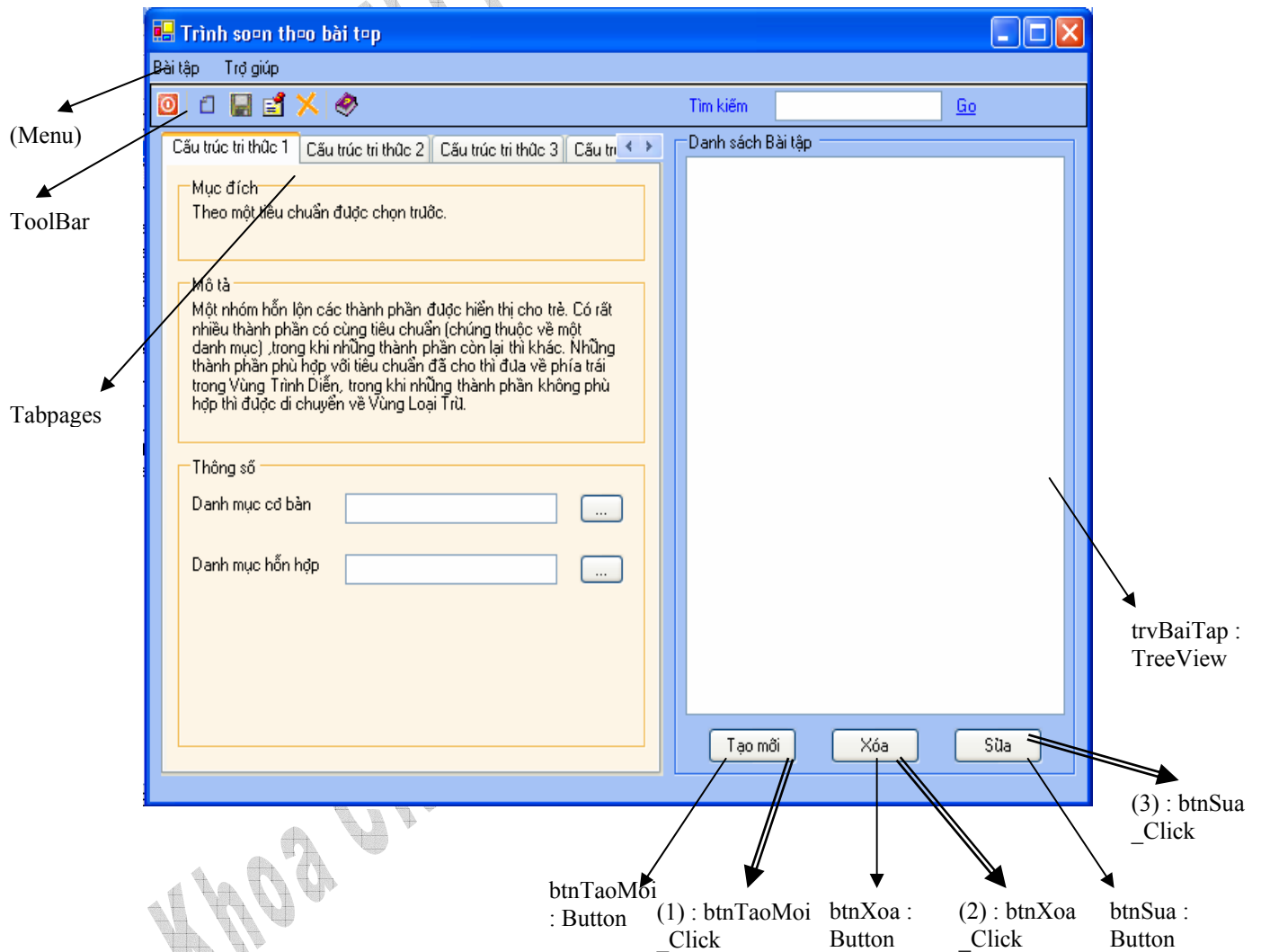
STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	ReadBitmap2ByteArray()	Bitmap image	byte[]	<pre> MemoryStream stream=new MemoryStream(); image.Save (stream,System.Drawing.Imaging.ImageFormat.Bmp); return stream.ToArray(); </pre>	chuyển đổi một ảnh ra mảng các byte
2	Tong()	Bitmap bit	int	<pre> int s=0; byte[] b=ReadBitmap2ByteArray(bit); for(int i=0;i&lt;b.Length;i++)     s+=b[i]; return s; </pre>	Tính tổng các byte của ảnh
3	SoSanh2DT()	Image m1,Image m2	string	<pre> string sHD=""; int t1=Tong((Bitmap)m1); int t2=Tong((Bitmap)m2); if(t1==t2) //2 doi tuong bang nhau {     sHD="Chính xác ! Chúc mừng em ! Hãy chú ý hình kế tiếp sắp xuất hiện."; } else { </pre>	Đưa ra hướng dẫn khi so sánh 2 ảnh



				<pre> sHD="Sai rồi ! Hãy chọn hình khác xem nào ! Nhanh lên kéo hết giờ đây !"; } return sHD; </pre>	
4	HuongDanThem()		string	<pre> return "Qua hình khác rồi ! Hãy quan sát và chọn nhanh lên nào !"; </pre>	Đưa ra hướng dẫn khi hết qua ảnh khác

**3.5/ Mô tả chi tiết lớp SOANTHAO :**

3.5.1/ Thiết kế giao diện :



## 3.5.1.1/ Thiết kế menu :

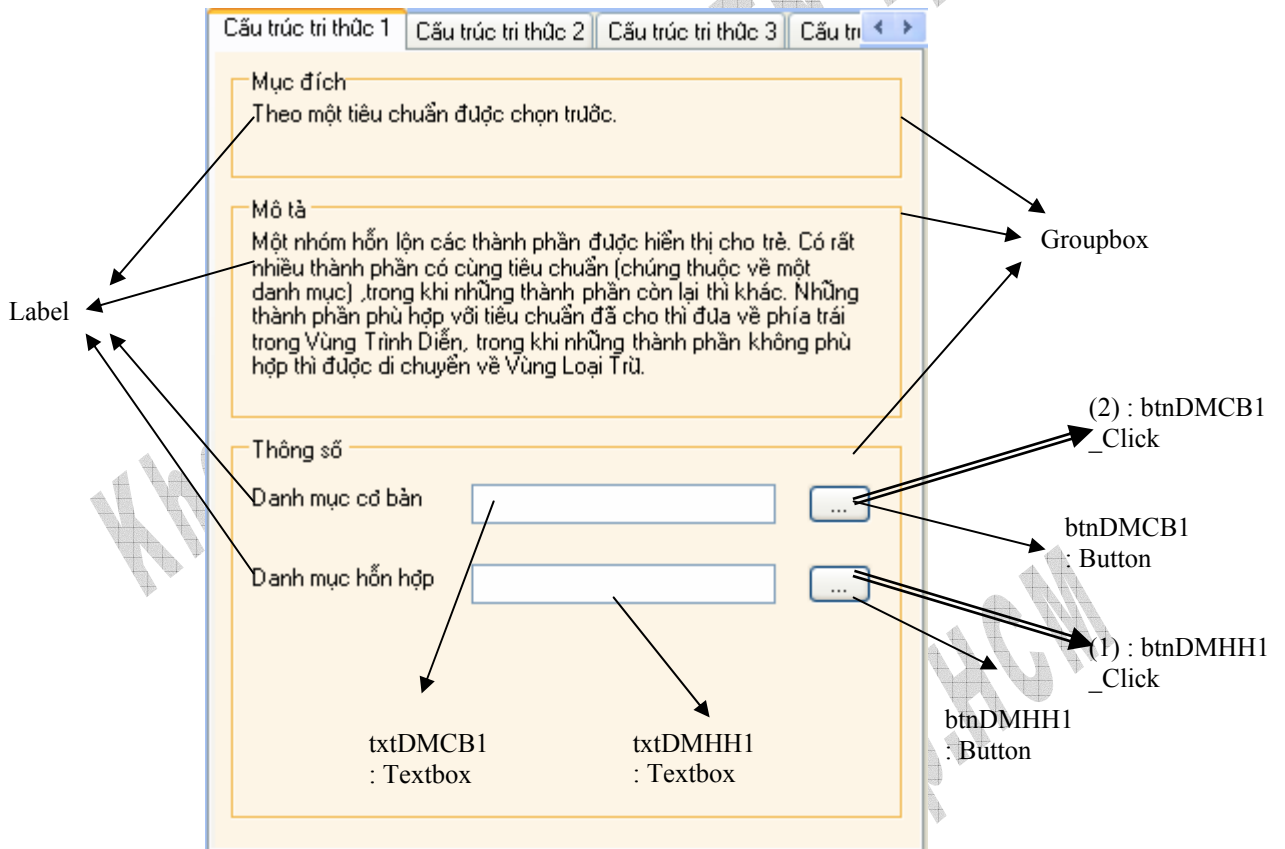
STT	Tên	Text	Shortcut	Ý nghĩa
1	mnuBaiTap	Bài tập		
2	mnuMoBT	Mở bài tập	Ctrl + O	Mở bài tập
3	mnuTaoBT	Tạo bài tập	Ctrl + C	Tạo bài tập mới đã soạn
4	mnuLuuBT	Lưu bài tập	Ctrl + S	Lưu bài tập đã tạo
5	mnuSuaBT	Sửa bài tập	Ctrl + R	Sửa bài tập đã tạo
6	mnuXoaBT	Xoá bài tập	Ctrl + D	Xoá bài tập đã tạo
7	mnuThoat	Thoát	Ctrl + X	Thoát khỏi màn hình
8	mnuTroGiup	Trợ giúp		
9	mnuHuongDan	Hướng dẫn		Hướng dẫn sử dụng phần mềm
10	mnuThongTin	Thông tin phần mềm		Thông tin về phần mềm

## 3.5.1.2/ Thiết kế Toolbar :

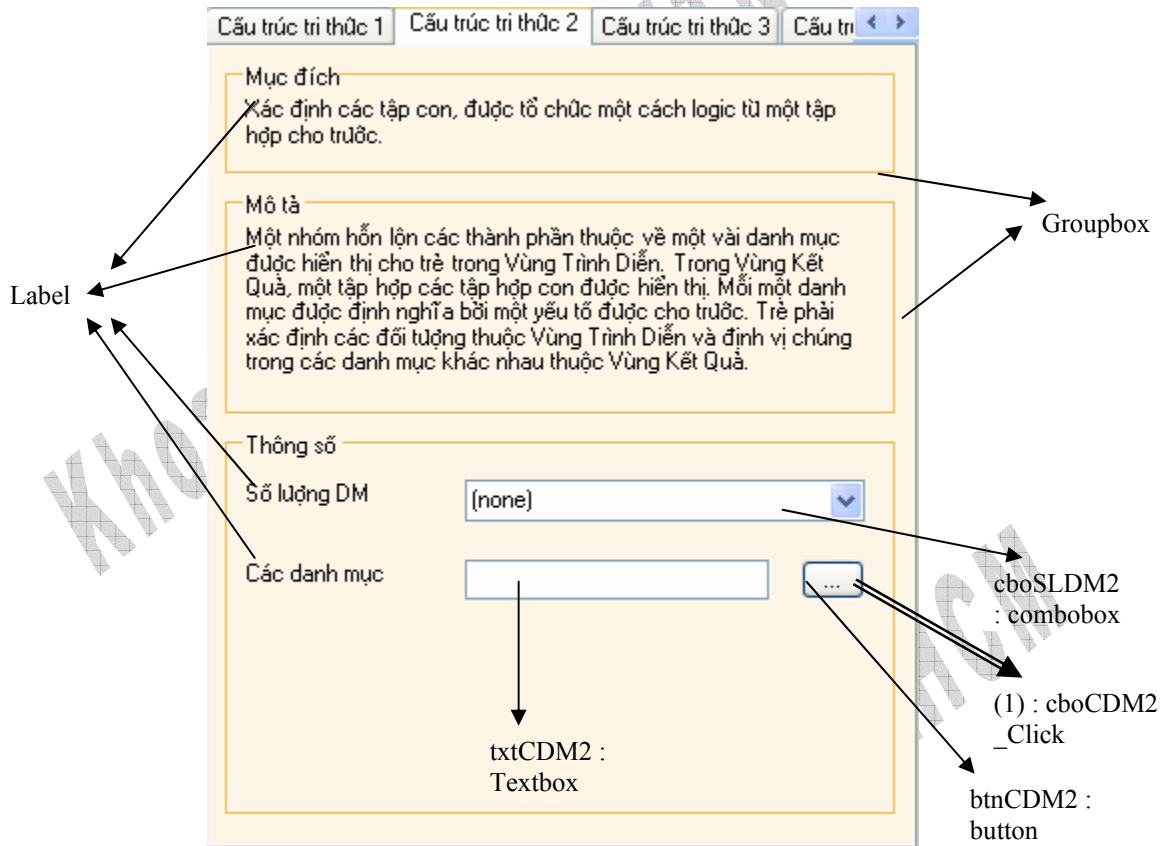
STT	Tên	Text	Ý nghĩa
1	toolBarThoat	(icon)	Thoát khỏi màn hình
2	toolBarTaoMoi	(icon)	Tạo mới bài tập
3	toolBarLuu	(icon)	Lưu bài tập đã tạo
4	toolBarSua	(icon)	Sửa bài tập
5	toolBarXoa	(icon)	Xoá bài tập đã tạo
6	toolBarHD	(icon)	Hướng dẫn sử dụng phần mềm

3.5.1.3/ Thiết kế các Tabpages:

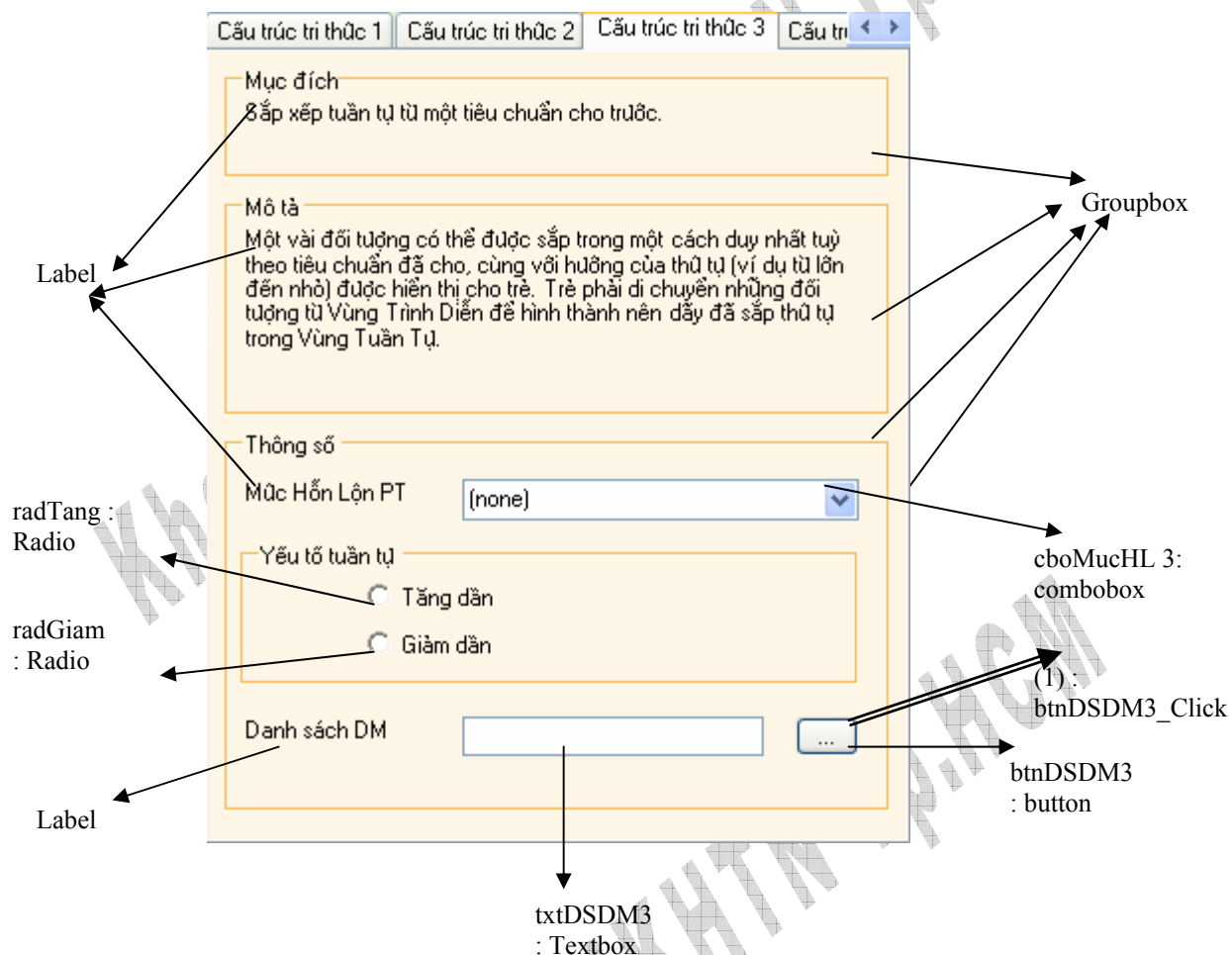
3.5.1.3.1/ Tabpage Cấu trúc tri thức 1 :



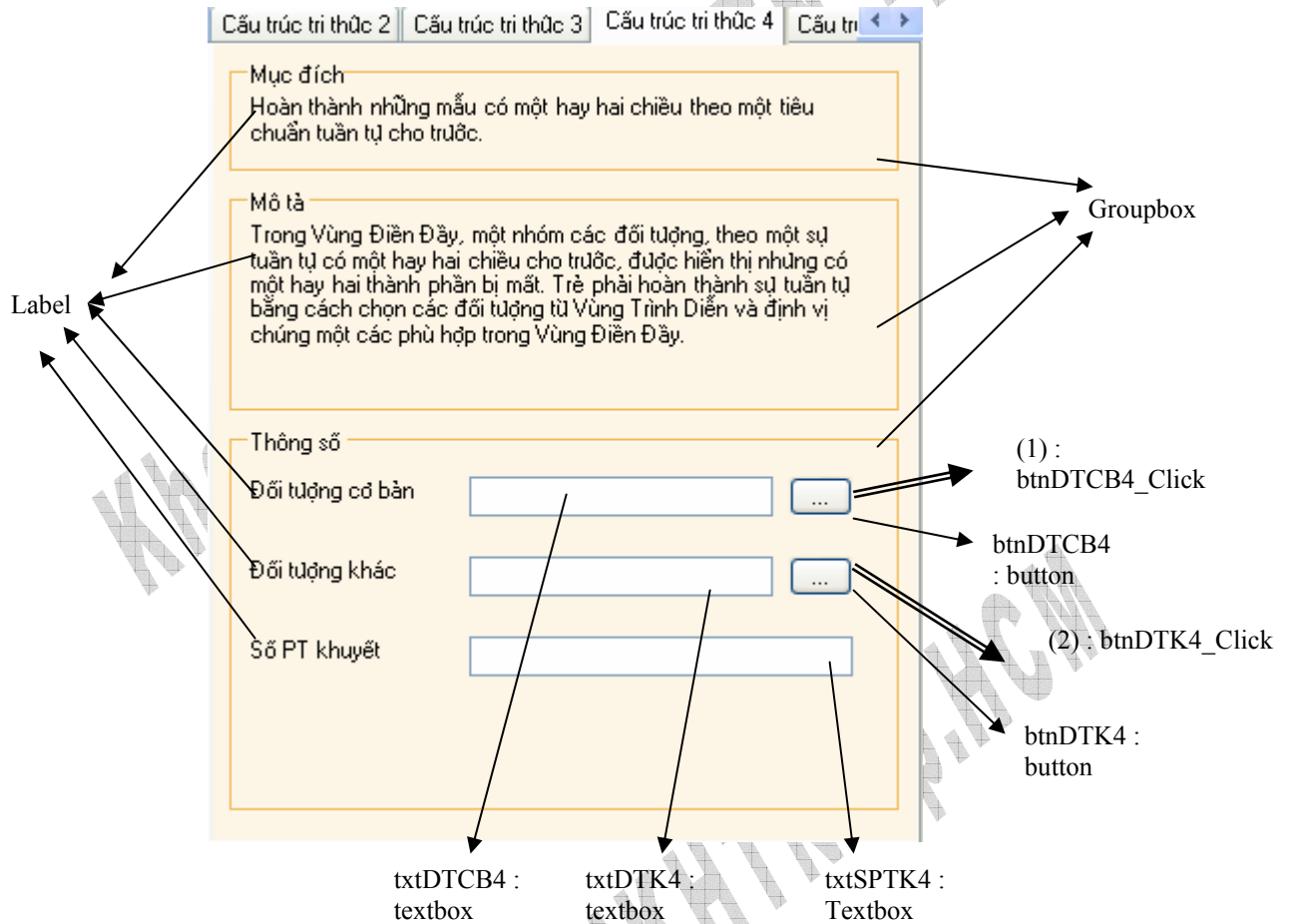
3.5.1.3.2/ Tabpage Cấu trúc tri thức 2 :



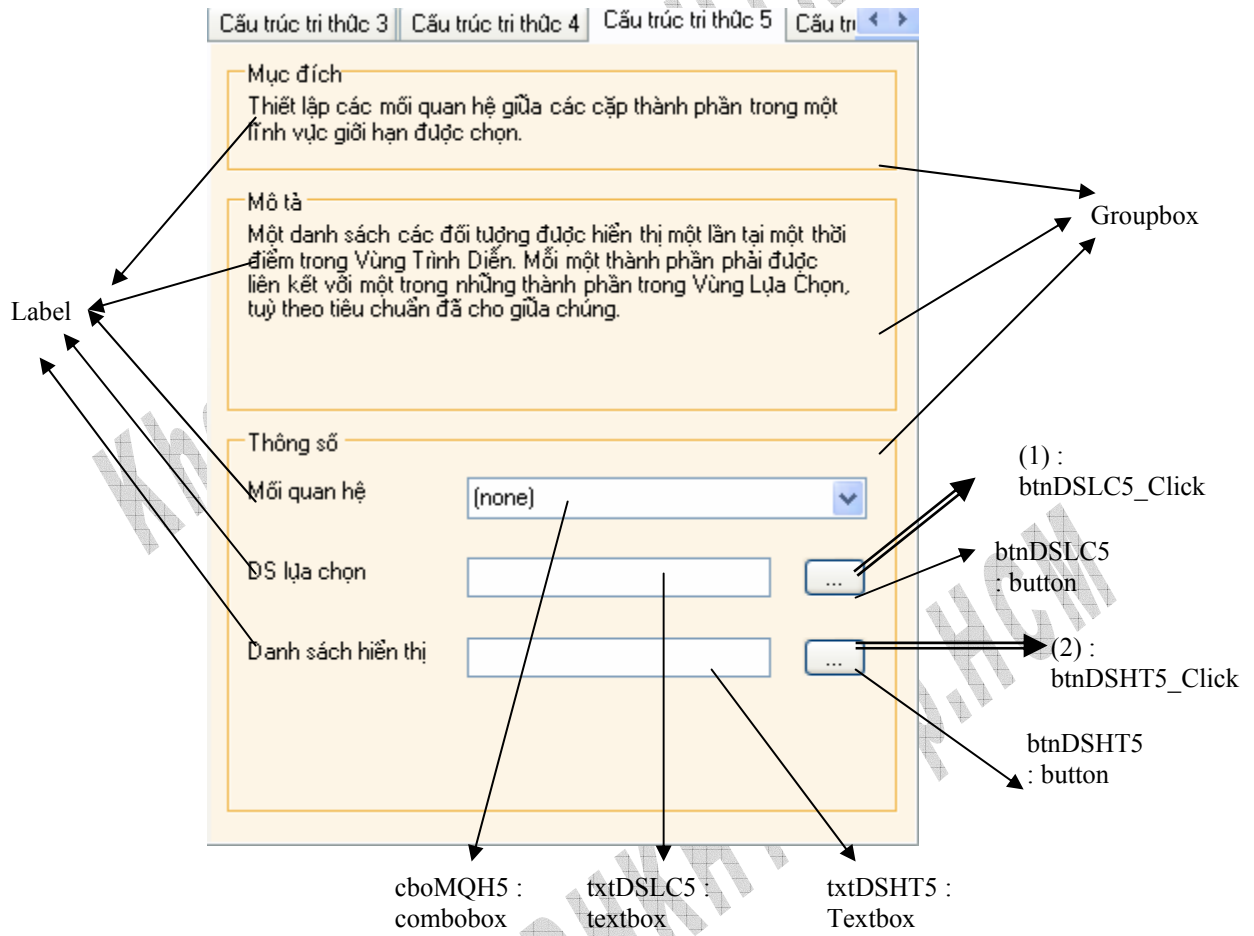
3.5.1.3.3/ Tabpage Cấu trúc tri thức 3 :



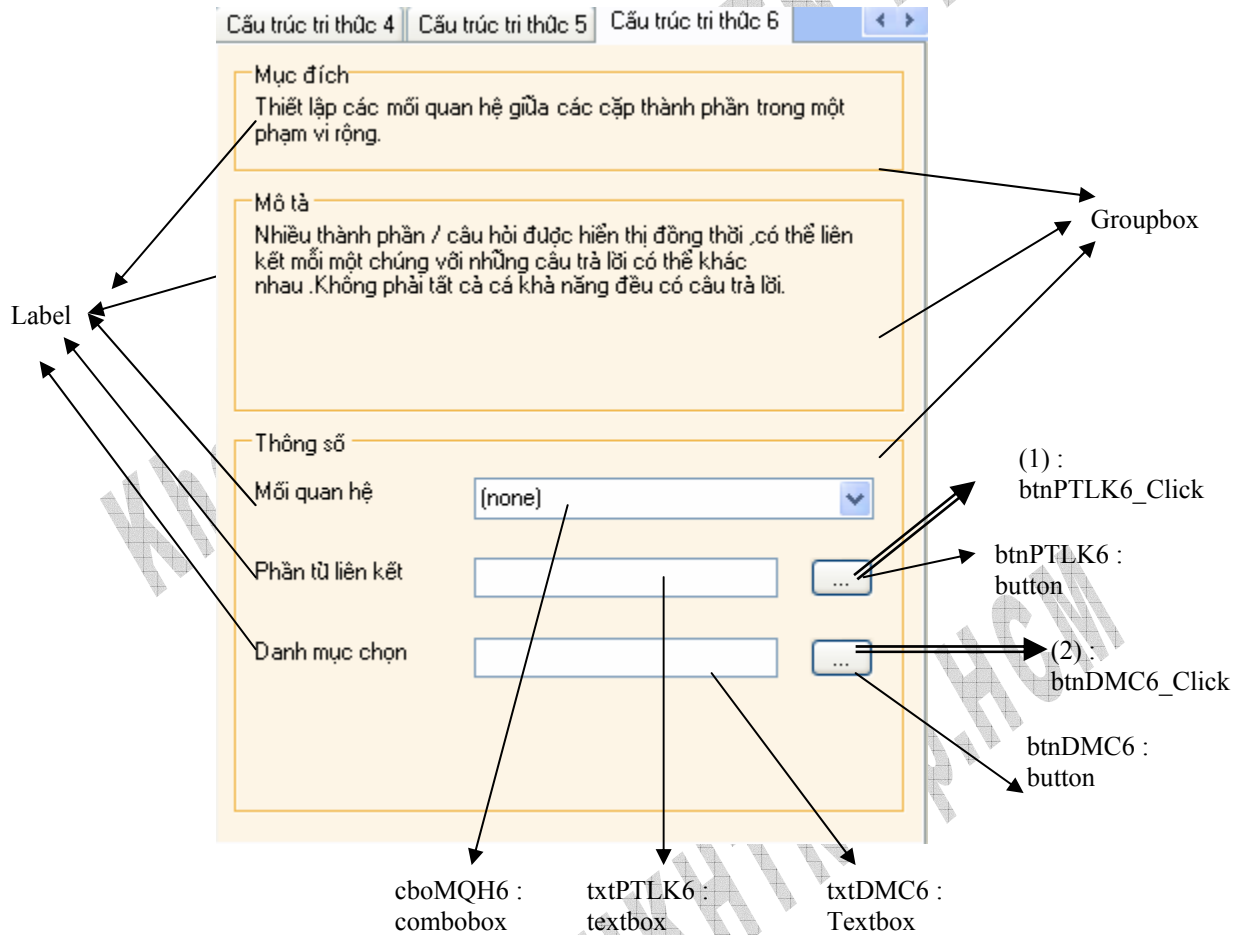
3.5.1.3.4/ Tabpage Cấu trúc tri thức 4 :



3.5.1.3.5/ Tabpage Cấu trúc tri thức 5 :



3.5.1.3.6/ Tabpage Cấu trúc tri thức 6 :



3.5.2/ Thiết kế biến và xử lý :

**Danh sách các đối tượng thể hiện**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	trvBaiTap	TreeView		
2	btnXoa	Button		
3	btnSua	Button		
4	btnTaoMoi	Button		
5	btnDSHT5	Button		



6	txtDSHT5	TextBox	
7	txtDMC6	TextBox	
8	txtDTK4	TextBox	
9	txtDTCB4	TextBox	
10	btnDTCB4	Button	
11	btnDMC6	Button	
12	btnCDM2	Button	
13	btnDTK4	Button	
14	txtSPTK4	TextBox	
15	radGiam	RadioButton	
16	radTang	RadioButton	
17	cboSLDM2	ComboBox	
18	txtCDM2	TextBox	
19	cboMQH6	ComboBox	
20	menuMoBT	MenuItem	
21	btnDSL5	Button	
22	txtDSL5	TextBox	
23	btnPTLK6	Button	
24	txtPTLK6	TextBox	
25	btnDSDM3	Button	
26	cboMQH5	ComboBox	
27	cboMucHL3	ComboBox	
28	txtDSDM3	TextBox	
29	txtDMCB1	TextBox	
30	btnDMHH1	Button	
31	txtDMHH1	TextBox	
32	btnDMCB1	Button	
33	tabControl_CTTT	TabControl	
34	tabPageCTTT1	TabPage	
35	tabPageCTTT2	TabPage	
36	tabPageCTTT3	TabPage	
37	tabPageCTTT4	TabPage	
38	tabPageCTTT5	TabPage	
39	tabPageCTTT6	TabPage	
40	toolBarThoat	ToolBarButton	
41	toolBarTaoTB	ToolBarButton	
42	toolBarLuuBT	ToolBarButton	
43	toolBarSuaBT	ToolBarButton	
44	toolBarXoaBT	ToolBarButton	
45	toolBarHD	ToolBarButton	
46	mainMenu1	MainMenu	
47	mnuBT	MenuItem	
48	mnuTaoBT	MenuItem	
49	mnuLuuBT	MenuItem	
50	mnuSuaBT	MenuItem	
51	mnuXoaBT	MenuItem	

52	mnuThoat	MenuItem		
53	mnuTG	MenuItem		
54	mnuTTPM_TG	MenuItem		
55	mnuHD_TG	MenuItem		
56	hlpPrvHelpChmFile	HelpProvider		

### Danh sách các biến thành phần

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	nDMCB1	int	Id danh mục cơ bản đã tạo	
2	nDMHH1	int	Id danh mục đã tạo	
3	TenDMCB	string	Tên danh mục đã tạo	

### Danh sách các sự kiện

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	btnTaoMoi_Click()	object sender, System.EventArgs e		switch(tabControl_CTTT.SelectedIndex) { case 0: TaoMoiBaiTap(ArrStr); ... }	
2	btnXoa_Click()	object sender, System.EventArgs e		... trvBaiTap.SelectedNode.Nodes. Remove(trvBaiTap.SelectedNode.FirstNode); ...	
3	btnSua_Click()	object sender, System.EventArgs e		... witch(tabControl_CTTT.SelectedIndex) { case 0: SuaBaiTap(ArrStr); break; }	
4	mnuThoat_Click()	object sender, EventArgs e		Application.Exit()	
5	mnuTaoBT_Click()	object sender, System.EventArgs e		btnTaoMoi_Click(this, EventArgs.Empty);	
6	mnuSuaBT_Click	object sender, System.EventArgs		btnSua_Click(this, EventArgs.Empty);	

		e			
7	mnuXoaBT_Click	object sender, System.EventArgs e		btnXoa_Click(this, EventArgs.Empty);	
8	mnuTTPM_TG_Click	object sender, System.EventArgs e		ThongtinPhanmem f = new ThongtinPhanmem(); f.ShowDialog();	
9	mnuHD_TG_Click	object sender, System.EventArgs e		Help.ShowHelp(this, @"Help\ITS.chm", HelpNavigator.TableOfContents);	
10	toolBar1_ButtonClick()	object sender, ToolBarButton ClickEventArgs e		if(e.Button.Equals(this.toolBarThoat)) { mnuThoat_Click(this, EventArgs.Empty); } ...	
11	OnDrawItem()	object sender, DrawItemEventArgs e		...	Hàm tùy biến menu
12	OnMeasureItem()	object sender, MeasureItemEventArgs e		...	Hàm tùy biến menu
13	OnDrawItemSec()	object sender, DrawItemEventArgs e		...	Hàm tùy biến menu
14	OnMeasureItemSec()	object sender, MeasureItemEventArgs e		...	Hàm tùy biến menu
15	OnDrawSpecial()	object sender, DrawItemEventArgs e		...	Hàm tùy biến menu
16	btnDMCB1_Click()	object sender, System.EventArgs e		DanhMuc dm = new DanhMuc(); dm.ShowDialog(); ...	
17	btnDMHH1_Click()	object sender, System.EventArgs e		DanhMuc dm = new DanhMuc(); dm.ShowDialog(); ...	
18	btnCDM2_Click()	object sender, System.EventArgs e		DanhMuc dm=new DanhMuc(); dm.ShowDialog(); ...	
19	btnDSDM3_Click	object sender, System.EventArgs e		DanhMuc dm=new DanhMuc(); dm.ShowDialog(); ...	
20	btnDTCB4_Click	object sender, System.EventArgs e		DanhMuc dm=new DanhMuc(); dm.ShowDialog(); ...	
21	btnDTK4_Click	object sender,		DanhMuc dm=new DanhMuc();	

		System.EventArgs e		dm.ShowDialog(); if(cboMQH5.SelectedIndex==0) { dm.ShowDialog(); ... } else if(cboMQH5.SelectedIndex==1) { dmam.ShowDialog(); ... } ... }	
22	btnDSL5_Click	object sender, System.EventArgs e		DanhMuc dm = new DanhMuc(); dm.ShowDialog(); ...	
23	btnDSHT5_Click	object sender, System.EventArgs e		DanhMuc dm = new DanhMuc(); DanhMucAmThanh dmam= new DanhMucAmThanh(); ...	
24	btnPTLK6_Click	object sender, System.EventArgs e		DanhMuc dm = new DanhMuc(); dm.ShowDialog(); ...	
25	btnDMC6_Click	object sender, System.EventArgs e		DanhMuc dm = new DanhMuc(); dm.ShowDialog(); ...	

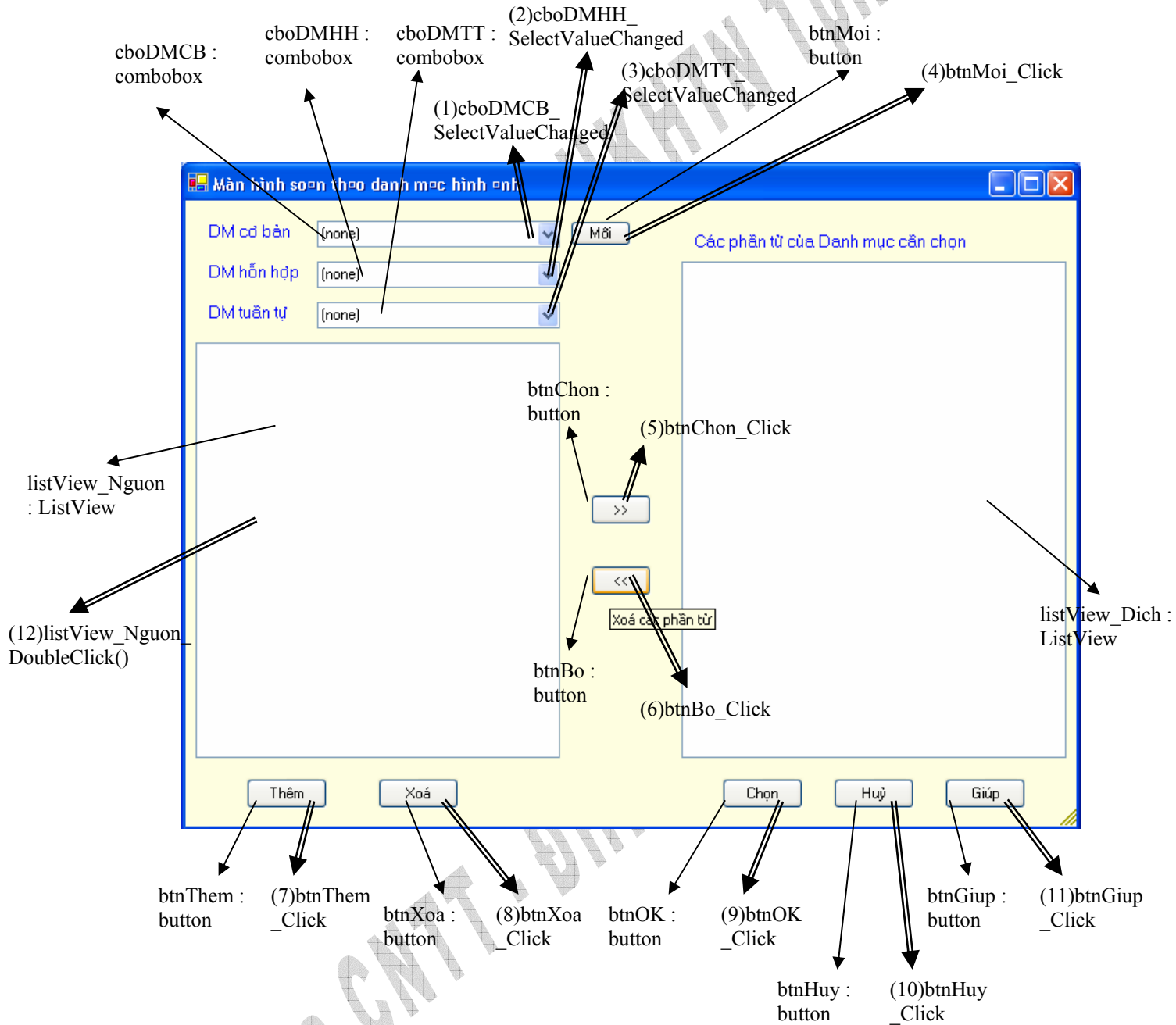
### Danh sách các hàm thành phần

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	SoanThao()				Hàm khởi tạo
2	TaoMoiBaiTap()	String[] ArrayStrNode	void	... for(int i=0;i<ArrayStrNode.Length;i++) ArrNode[i]=new TreeNode(ArrayStrNode[i],1,1); ...	Hàm tạo các bài tập mới trong cây bài tập
3	SuaBaiTap()	String[] ArrayStrNode		... for(int i=0;i<ArrayStrNode.Length;i++) ArrNode[i]=new TreeNode(ArrayStrNode[i],1,1); ...	Hàm sửa mới bài tập trong cây bài tập
4	ThemBaiTap_C1()			... ltdl.InsertBaitap_C1(slBaitap, TenDMCB,nDMCB1,nDMHH1); ...	Thêm bài tập vào CSDL

5	SuaTenDMCB()	ref string s		<pre>s.Trim(); for(int i=0;i&lt;s.Length;i++)     if(s[i].ToString()==" ")     {         s=s.Substring(0,i+1);         break;     }</pre>	Lấy tên đầu của chuỗi
6	ThemBaiTap_C2()			<pre>... ltdl.InsertBaitap_C2(slBaitap, TenDMCB,int.Parse(cboSLDM2.Text), nDMCB1); ...</pre>	Thêm bài tập vào CSDL
7	ThemBaiTap_C3()			<pre>... ltdl.InsertBaitap_C3(slBaitap,"bai tap " + slBaitap.ToString(), (byte)cboMucHL3.SelectedIndex, (byte)bcheck,nDMCB1); ...</pre>	Thêm bài tập vào CSDL
8	ThemBaiTap_C4()			<pre>... ltdl.InsertBaitap_C4(slBaitap,"bai tap " + slBaitap.ToString(),nDMCB1,nDMHH1, int.Parse(txtSPTK4.Text)); ...</pre>	Thêm bài tập vào CSDL
9	ThemBaiTap_C5()			<pre>... ltdl.InsertBaitap_C5(slBaitap,"bai tap " + slBaitap.ToString(),nDMCB1,nDMHH1 ,cboMQH5.SelectedIndex); ...</pre>	Thêm bài tập vào CSDL
10	ThemBaiTap_C6()			<pre>... ltdl.InsertBaitap_C6(slBaitap,"bai tap " + slBaitap.ToString(),nDMCB1,nDMHH1, cboMQH6.SelectedIndex); ...</pre>	Thêm bài tập vào CSDL

**3.6/ Mô tả chi tiết lớp DANHMUC :**

3.6.1/ Thiết kế giao diện :



## 3.6.2/ Thiết kế biến và xử lý :

**Danh sách các đối tượng thể hiện**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	listViewNguon	ListViewXPDrpDrop		
2	listViewDich	ListViewXPDrpDrop		
3	btnThem	Button		
4	btnGiup	Button		
5	btnBo	Button		
6	btnHuy	Button		
7	btnChon	Button		
8	btnXoa	Button		
9	cboDMTT	ComboBox		
10	btnMoi	Button		
11	cboDMHH	ComboBox		
12	cboDMCB	ComboBox		
13	btnOK	Button		

**Danh sách các biến thành phần**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	nImages	int	số lượng ảnh có	
2	nDanhMuc		số lượng danh mục	
3	bChanged	int	biến kiểm tra danh mục nào đang được chọn	
4	sDMCB	string	Tên chuỗi danh mục	
5	image	Bitmap	ảnh	
6	xllv	XuLy_ListView	lớp xử lý listview	

**Danh sách các sự kiện**

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	btnOK_Click()	object sender, System.EventArgs e		... LuuDanhMuc(); ... this.DialogResult = DialogResult.OK; this.Close();	
2	btnMoi_Click	(nt)		ThemDanhMuc dm = new ThemDanhMuc(); dm.ShowDialog();	

				...	
3	btnChon_Click	(nt)		<pre> ... listViewDich.SetExStyles(); if(bChanged==0) {     switch(cboDMCB.SelectedIndex)     {         case 0 : /             xllv.ThemPTImageList(imageList_Ca ,listViewNguon,imageList_Dich);             break;         ...     } </pre>	
4	btnBo_Click	(nt)		<pre> xllv.XoaPhanTu(listViewDich,imageList_Dic h); </pre>	
5	btnHuy_Click	(nt)		<pre> this.DialogResult = DialogResult.Cancel; this.Close(); </pre>	
6	btnThem_Click	(nt)		<pre> OpenFileDialog fdl = new OpenFileDialog(); filpath = fdl.FileName; System.Drawing.Image myImage=Image.FromFile(filpath); if(bChanged==0) {     switch(cboDMCB.SelectedIndex)     {         case 0 :             imageList_Ca.Images.Add(myImage) ;             break;         ...     } </pre>	
7	btnXoa_Click	(nt)		<pre> if(bChanged==0) {     switch(cboDMCB.SelectedIndex)     {         case 0 :             xllv.XoaPhanTu(listViewNguon,imag eList_Ca);             break;         ...     } </pre>	
8	cboDMCB_SelectedVa lueChanged()	object sender, EventArgs e		<pre> ... switch(cboDMCB.SelectedIndex) {     case 0 :         listViewNguon.LargeImageList = imageList_Ca; </pre>	



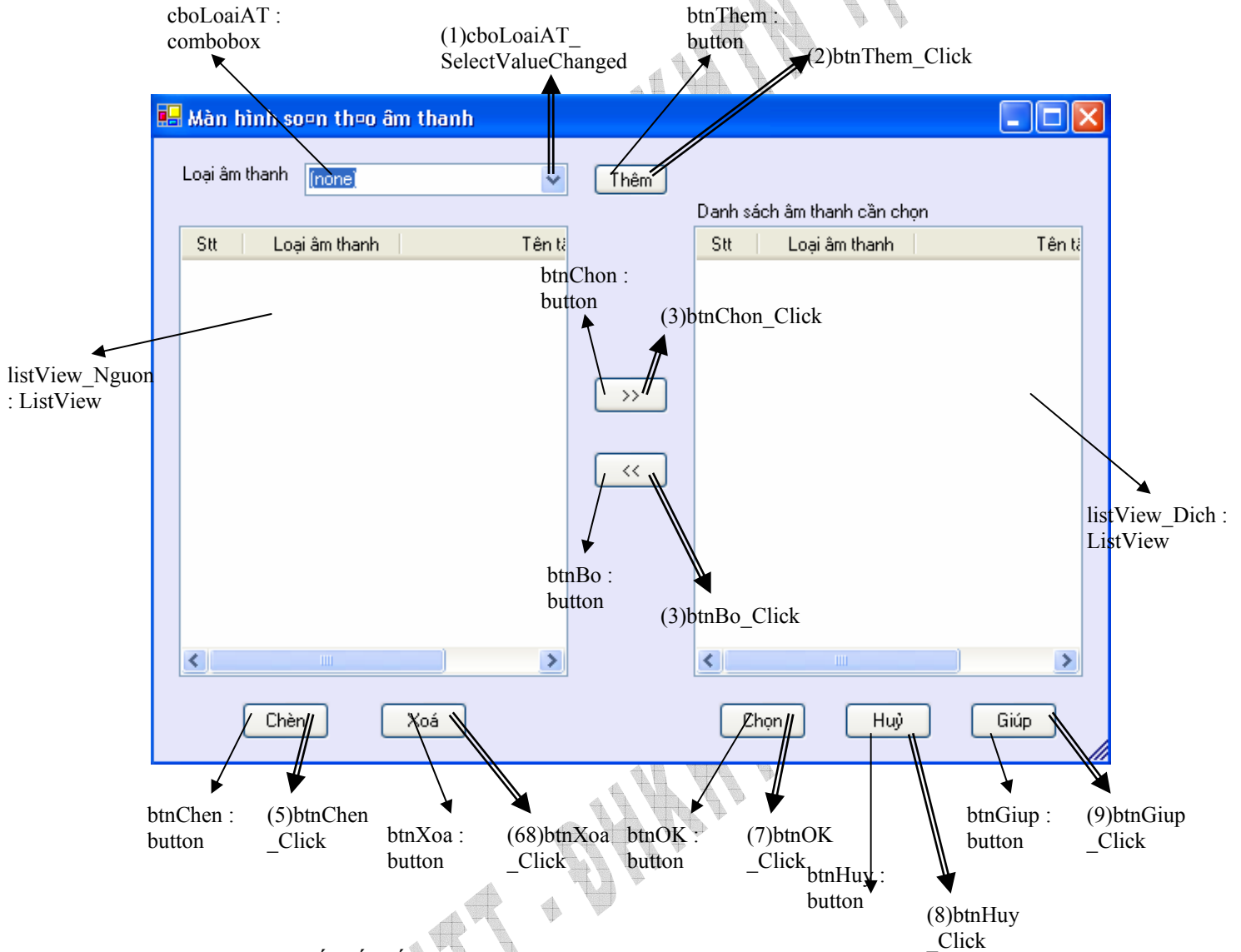
				<pre> nImages = imageList_Ca.Images.Count; break; ... } </pre>	
9	cboDMHH_SelectedValueChanged	(nt)		<pre> switch(cboDMHH.SelectedIndex) { case 0 : listViewNguon.LargeImageList = imageList_All; nImages = imageList_All.Images.Count; break; ... } </pre>	
10	listViewNguon_DoubleClick	(nt)		<pre> ListViewItem focusitem = listViewNguon.FocusedItem; if(focusitem != null) { if(bChanged==0) { switch(cboDMCB.SelectedIndex) { case 0 : image =new Bitmap(imageList_Ca.Images[focusitem. ImageIndex],200,200); break; ... } } } </pre>	
11	cboDMTT_SelectedValueChanged	(nt)		<pre> ... switch(cboDMTT.SelectedIndex) { case 0 : listViewNguon.LargeImageList = imageList_TT1; nImages = imageList_TT1.Images.Count; break; ... } </pre>	
12	btnGiup_Click()	object sender, System.EventArgs e		<pre> Help.ShowHelp(this,@"Help\ITS.chm",Help Navigator.Index); </pre>	

### Danh sách các hàm thành phần

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	DanhMuc()			bChanged=-1; xllv = new XuLy_ListView();	khởi tạo lớp
2	LayTenDM()		String	return sDMCB;	Lấy tên danh mục
3	DaCoPhanTu()		bool	if(listViewDich.Items.Count==0) return false; return true;	Kiểm tra xem đã có phần tử trong listview đích hay chưa ?
4	ReadBitmap2ByteArray()	Bitmap image	byte[]	MemoryStream stream=new MemoryStream(); image.Save(stream,System.Drawing. Imaging.ImageFormat.Bmp); return stream.ToArray();	chuyển đổi ảnh thành các byte
5	ReadByteArrayToBitmap()	byte[] content	Bitmap	MemoryStream stream = new MemoryStream(content); Bitmap image = new Bitmap(stream); return image;	chuyển đổi các byte thành ảnh
6	LuuDanhMuc()			... tdl.InsertDanhMuc(nDanhMuc,"Danh muc " + nDanhMuc.ToString(),"Hinh anh"); ... byte[] b=ReadBitmap2ByteArray((Bitmap) imageList_Dich.Images[i]); ltdl.InsertImage(nPhoto+i,"h" + (nPhoto+i).ToString(),ref b,nDanhMuc); ...	Lưu danh mục đã chọn ( cùng với nội dung của nó – hình )
7	Lay_DanhMuc_ID()		int	return nDanhMuc;	Lấy id của danh mục
8					

**3.7/ Mô tả chi tiết lớp DANHMUCAMTHANH :**

3.7.1/ Thiết kế giao diện :



3.7.2/ Thiết kế biến và xử lý :

**Danh sách các đối tượng thể hiện**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	cboLoaiAmThanh	ComboBox		
2	listView_Nguồn	ListViewXPDrappDrop		

3	listView_Dich	ListViewXPDrpDrop		
4	btnXoa	Button		
5	btnChen	Button		
6	btnThem	Button		
7	btnCh	Button		
8	btnBo	Button		
9	btnHuy	Button		
10	btnChon	Button		
11	btnGiup	Button		
12				

### Danh sách các biến thành phần

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	nDM	int	id của danh mục đã lưu	
2				

### Danh sách các sự kiện

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	btnThem_Click()	object sender, System.EventArgs e		ThemDanhMuc lam = new ThemDanhMuc(); ... lam.ShowDialog(); ... cboLoaiAmThanh.Items.Add((object)strLoai); ; ...	
2	btnCh_Click()	object sender, System.EventArgs e		ThemListViewDich();	
3	btnBo_Click()	object sender, System.EventArgs e		XoaPhanTuDich();	
4	btnChen_Click	object sender, System.EventArgs e		OpenFileDialog fdl= new OpenFileDialog(); ... ThemPhanTu(listView_Nguon,cboLoaiAmThanh.Text,fpath); ...	
5	btnXoa_Click	object sender, System.EventArgs e		... listView_Nguon.Items.RemoveAt(n); CapNhat_ListView(listView_Nguon,n); ...	

6	btnChon_Click	object sender, System.EventArgs e		<pre>if(DaCoPhanTu()) {     LuuAmThanh();     this.DialogResult = DialogResult.OK;     this.Close(); } ...</pre>	
7	btnHuy_Click	object sender, System.EventArgs e		<pre>this.DialogResult = DialogResult.Cancel; this.Close();</pre>	
8	cboLoaiAmThanh_SelectedValueChanged()	object sender, EventArgs e		<pre>switch(cboLoaiAmThanh.Text) {     case "Động vật":         ThemPhanTu(listView_Nguon,cboLoaiAmThanh.Text,s + @"\amthanhchim\bwd1.wav");         ...         break; } ...</pre>	
9	btnGiup_Click()	object sender, System.EventArgs e		<pre>Help.ShowHelp(this,@"Help\ITS.chm",HelpNavigator.KeywordIndex);</pre>	

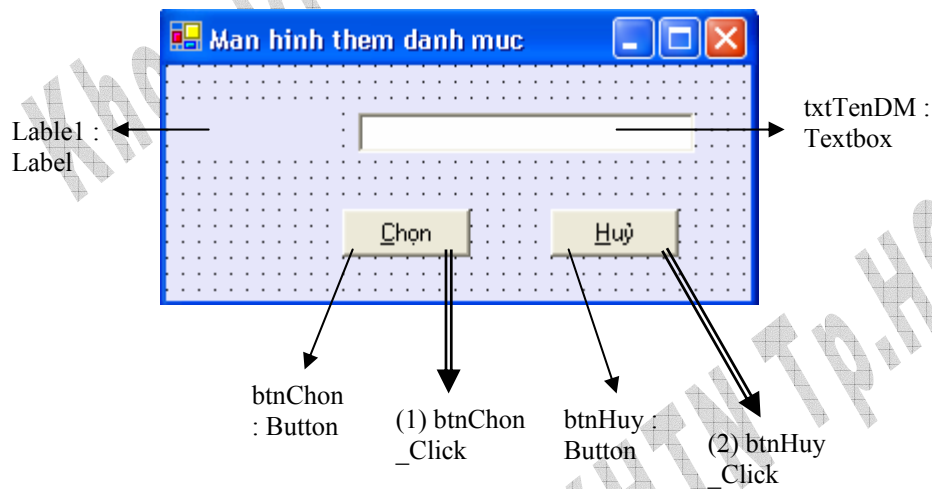
### Danh sách các hàm thành phần

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	ThemListViewDich()			<pre>... listView_Dich.Items.Add(newitem);</pre>	Thêm các phần tử vào listview đích
2	XoaPhanTuDich()			<pre>... listView_Dich.Items.Remove(item); ... CapNhat_ListView(listView_Dich,i); ...</pre>	Xoá một phần tử trong listview đích
3	ThemPhanTu()	ListView lv,string s2,string s3		<pre>... ListViewItem newitem = new ListViewItem(new string[] {stt.ToString(),s2,s3}); lv.Items.Add(newitem); ...</pre>	Thêm một phần tử vào listview
4	CapNhat_ListView()	ListView lv,int nItem		<pre>for(int i=nItem; i&lt;lv.Items.Count;i++) {     lv.Items[i].Text=(i+1).ToString(); }</pre>	Cập nhật lại listview sau khi xử lý
5	LuuAmThanh()			<pre>...</pre>	Lưu lại âm

				ltdl.InsertSound(nSounds+i, loai,duongdan,nDM); ... return nDM;	thanh đã chọn
6	Lay_DanhMuc_ID()	int		return nDM;	Lấy id của danh mục
7	DaCoPhanTu()	bool		if(listView_Dich.Items.Count!=0) return true; return false;	Kiểm tra xem đã có phần tử ở listview đích hay chưa ?
8					

**3.8/ Mô tả chi tiết lớp THEMDANHMUC :**

3.8.1/ Thiết kế giao diện :



3.8.2/ Thiết kế biến và xử lý :

**Danh sách các đối tượng thể hiện**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	label1	Label		
2	btnChon	Button		
3	btnHuy	Button		
4	txtDanhMuc	TextBox		

**Danh sách các biến thành phần**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1				
2				

### Danh sách các sự kiện

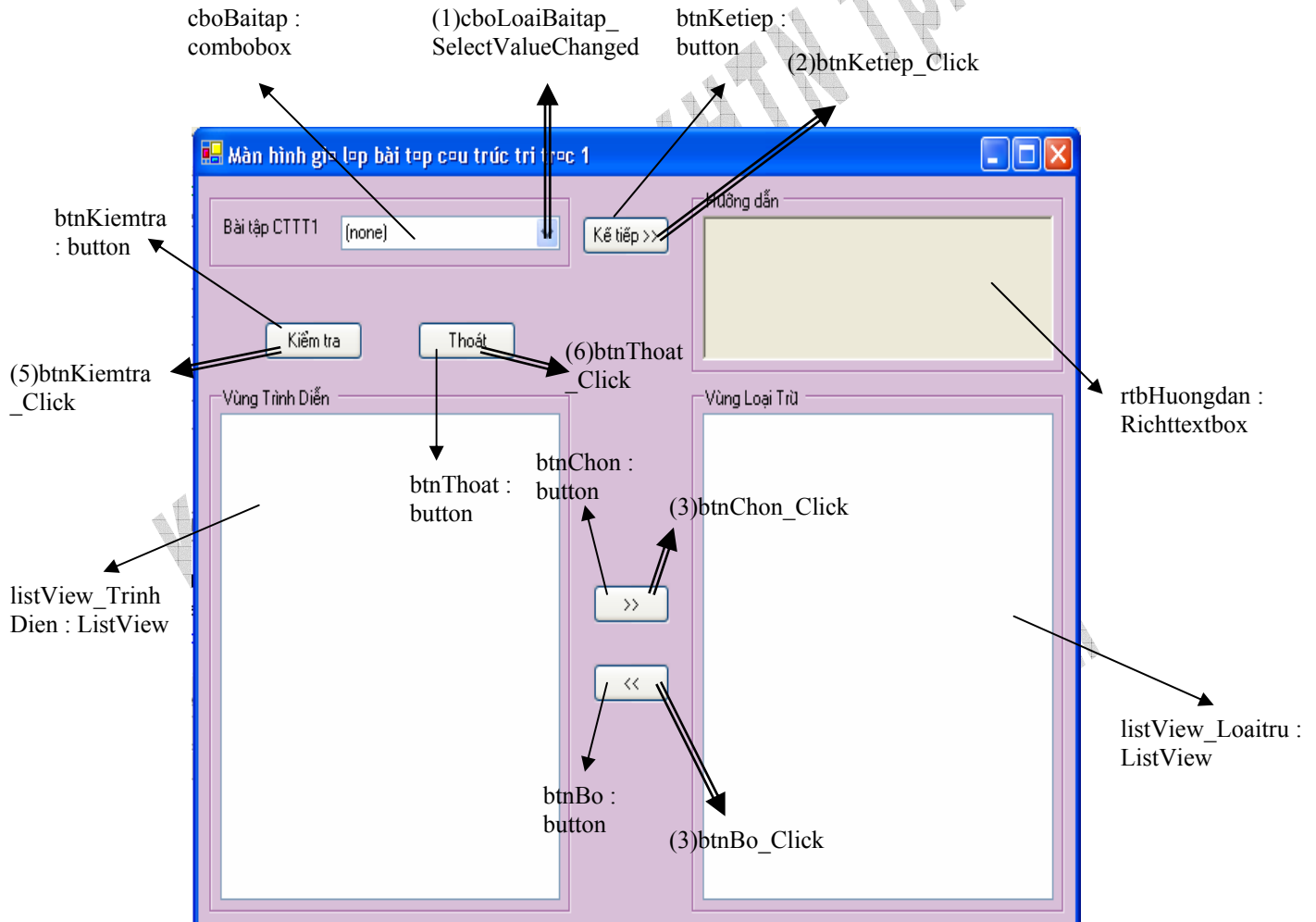
STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	btnChon_Click()	object sender, System.EventArgs e		this.DialogResult = DialogResult.OK; this.Close();	
2	btnHuy_Click	object sender, System.EventArgs e		this.DialogResult = DialogResult.Cancel; this.Close();	

### Danh sách các hàm thành phần

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	LayTen()		string	return txtDanhMuc.Text;	Lấy tên danh mục muốn tạo
2					

**3.9/ Mô tả chi tiết lớp LAMBAITAP C1 :**

3.9.1/ Thiết kế giao diện :



3.9.2/ Thiết kế biên và xử lý :

**Danh sách các đối tượng thể hiện**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	cboBaitap	ComboBox		
2	listView_DMCB	ListViewXPDrappDrop		
3	listView_DMHH	ListViewXPDrappDrop		
4	btnChon	Button		
5	btnBo	Button		
6	btnKiemtra	Button		
7	btnThoat	Button		



8	btnKetiep	Button	
9	rtbHuongDan	RichTextBox	

### Danh sách các biến thành phần

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	ltdl	LT_DuLieu		
2	xllv	XuLy_ListView		
3	image	Bitmap		

### Danh sách các sự kiện

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	LamBaiTap_Closing()	object sender, CancelEventArgs e		if(ltdl.IsConnOpen()) ltdl.DongKetNoi();	
2	btnThoat_Click()	object sender, EventArgs e		if(ltdl.IsConnOpen()) ltdl.DongKetNoi(); this.Close();	
3	cboBaitap_SelectedValueChanged()	object sender, EventArgs e		... rtbHuongDan.Text = s1+ cboBaitap.Text + s2; DienDayDMCB(i+1); DienDayDMHH(i+1); ...	
4	btnChon_Click	object sender, System.EventArgs e		xllv.ThemPTImageList(imageList_DMCCB,listView_DMCCB,imageList_DMHH); xllv.ThemPTListView(listView_DMHH,imageList_DMHH); xllv.XoaPhanTu(listView_DMCCB,imageList_DMCCB);	
5	btnBo_Click()	object sender, System.EventArgs e		xllv.ThemPTImageList(imageList_DMHH,listView_DMHH,imageList_DMCCB); xllv.ThemPTListView(listView_DMCCB,imageList_DMCCB); xllv.XoaPhanTu(listView_DMHH,imageList_DMHH);	
6	listView_DMHH_DoubleClick()	object sender, EventArgs e		image =new Bitmap(imageList_DMHH.Images[focusitem.ImageIndex],200,200); HienThiAnh hta= new HienThiAnh(); hta.pbxAnh.Image = image; hta.ShowDialog(this);	
7	listView_DMCCB_DoubleClick()	object sender, EventArgs e		image =new Bitmap(imageList_DMCCB.Images[focusitem.	

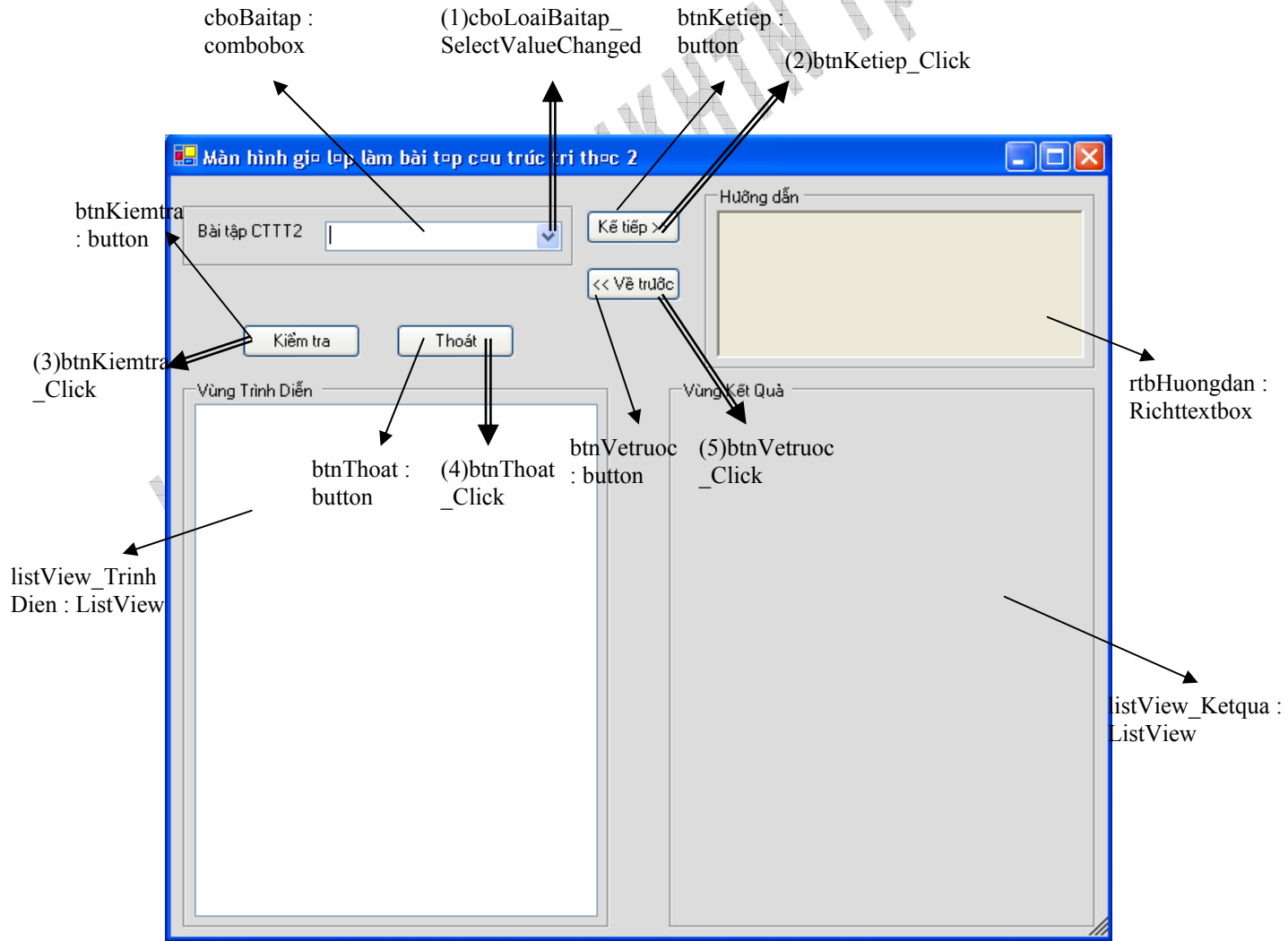
				ImageIndex],200,200); HienThiAnh hta= new HienThiAnh(); hta.pbxAnh.Image = image; hta.ShowDialog(this);	
8	btnKetiep_Click()	object sender, System.EventArgs e		LamBaiTap_C2 fbt = new LamBaiTap_C2(); this.Hide(); fbt.ShowDialog();	

### Danh sách các hàm thành phần

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	LamBaiTap_C1()			ltdl = new LT_DuLieu(); ltdl.MoKetNoi(); DienDayBaiTap(); xllv = new XuLy_ListView();	Hàm khởi tạo
2	ReadByteArrayToBitmap()	byte[] content	Bitmap	MemoryStream stream = new MemoryStream(content); Bitmap image = new Bitmap(stream); return image;	Hàm chuyển đổi các byte thành ảnh
3	DienDayDMCB()	int baitap_id		... dtreader = ltdl.LayDMCB1(baitap_id); ... xllv.Them1PT(listView_DM CB, imageList_DM CB.Images.Count-1); ....	Lấy danh mục cơ bản của bài tập từ CSDL
4	DienDayDMHH()	int baitap_id		... dtreader = ltdl.LayDMHH1(baitap_id); ... xllv.Them1PT(listView_DM HH, imageList_DM HH.Images.Count-1); ...	Lấy danh mục cơ bản của bài tập từ CSDL
5	DienDayBaiTap()			... dtreader = ltdl.LayBaiTapCTTT1()); ... cboBaitap.Items.Add (dtreader.GetSqlString(1)); ...	lấy tất cả bài tập của dạng 1
6					

**3.10/ Mô tả chi tiết lớp LAMBAITAP C2 :**

3.10.1/ Thiết kế giao diện :



## 3.10.2/ Thiết kế biến và xử lý :

**Danh sách các đối tượng thể hiện**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	btnThoat	Button		
2	btnKiemtra	Button		
3	listView Trinhdien	ListViewXPDrpDrop		
4	rtbHuongdan	RichTextBox		
5	btnKetiep	Button		
6	cboBaitap	ComboBox		
7	btnVetruoc	Button		

**Danh sách các biến thành phần**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	m_image	Image		
2	xllv	XuLy_ListView		
3	ArrImlI	ImageList[]	mảng các Imagelist	
4	ArrLv	ListViewXPDrpDrop[]	mảng các listview	
5	nKq	int	số danh mục	
6	image	Bitmap		

**Danh sách các sự kiện**

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	btnKetiep_Click()	object sender, System.EventArgs e		LamBaiTap_C3 fbt = new LamBaiTap_C3(); this.Hide(); fbt.ShowDialog();	
2	btnThoat_Click	object sender, System.EventArgs e		if(ltdl.IsConnOpen()) ltdl.DongKetNoi(); this.Close(); Application.Exit();	
3	LamBaiTap_C2_Closing()	object sender, CancelEventArgs e		if(ltdl.IsConnOpen()) ltdl.DongKetNoi(); this.Close(); Application.Exit();	
4	cboBaitap_SelectedValueChanged()	object sender, EventArgs e		... this.rtbHuongdan.Text = s1 + cboBaitap.Text + s2; TaoNoiDungBT(i+1);	

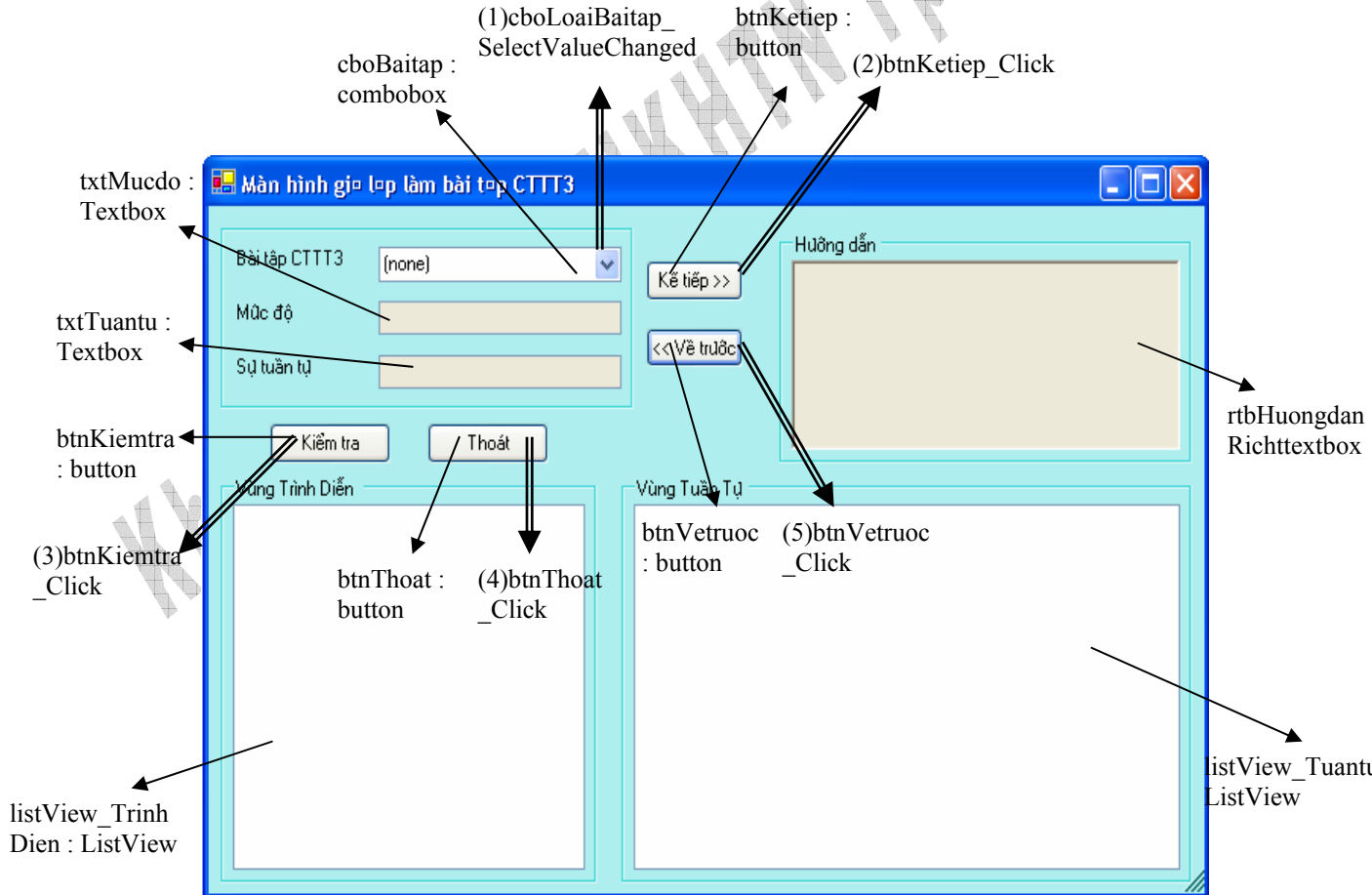
				...	
5	btnVetruoc_Click()	object sender, EventArgs e		LamBaiTap_C1 fbt = new LamBaiTap_C1(); this.Hide(); fbt.ShowDialog();	
6	listView_Trinhdien_DoubleClick()	object sender, EventArgs e		image = new Bitmap(imageList_Trinhdien.Images[focuse m.ImageIndex],200,200); HienThiAnh hta= new HienThiAnh(); hta.pbxAnh.Image = image; hta.ShowDialog(this);	
7					

### Danh sách các hàm thành phần

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	LamBaiTap_C2()			ltdl = new LT_DuLieu(); ltdl.MoKetNoi(); DienDayBaiTap(); xllv = new XuLy_ListView();	Hàm khởi tạo
2	ReadByteArrayToBitmap()	byte[] content	Bitmap	MemoryStream stream = new MemoryStream(content); Bitmap image = new Bitmap(stream); return image;	Hàm chuyển đổi các byte thành ảnh
3	DienDayVungTrinhDien()	int baitap_id		... dtreader = ltdl.LayDanhMucC2(baitap_id); ... xllv.Them1PT(listView_Trinhdien, imageList_Trinhdien.Images.Count-1);	Lấy danh mục trình diễn của bài tập
4	DienDayBaiTap()			... dtreader = ltdl.LayBaiTapCTTT2(); ... cboBaitap.Items.Add (dtreader.GetSqlString(1));	Lấy tất cả bài tập dạng 2
5	TaoNoiDungBT()	int baitap		DienDayVungTrinhDien(baitap); ... for(int j=0;j<nKq;j++) { ... ArrLv[j]= new ListViewXPDrápAndDrop. ListViewXPDrápDrop(); ... }	Tạo nội dung bài tập hiển thị và kết quả
6					

**3.11/ Mô tả chi tiết lớp LAMBAITAP C3 :**

3.11.1/ Thiết kế giao diện :



3.11.2/ Thiết kế biến và xử lý :

**Danh sách các đối tượng thể hiện**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	rtbHuongdan	RichTextBox		
2	btnVetruoc	Button		
3	btnKiemtra	Button		
4	btnThoat	Button		
5	txtMucdo	TextBox		
6	txtTuantu	TextBox		

7	listView_Tuantu	ListViewXPDrpDrop		
8	listView_Trinhdien	ListViewXPDrpDrop		
9	btnKetiep	Button		
10	cboBaitap	ComboBox		
11				

### Danh sách các biến thành phần

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	ltdl	LT_DuLieu	biến dữ liệu	
2	image	Bitmap	ảnh	
3	xllv	XuLy ListView		
4	m_nDoituongTruoc	Image	ảnh đã di chuyển trước đó	
5	m_nDoituongHientai	Image	ảnh đang di chuyển	
6	m_nTuantu	byte	yếu tố tuần tự	
7	hd3	HuongDan_C3	lớp hướng dẫn	
8				

### Danh sách các sự kiện

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	btnKetiep_Click()	object sender, System.EventArgs e		LamBaiTap_C4 fbt = new LamBaiTap_C4(); this.Hide(); fbt.ShowDialog();	
2	btnThoat_Click()	object sender, System.EventArgs e		if(ltdl.IsConnOpen()) ltdl.DongKetNoi(); this.Close(); Application.Exit();	
3	LamBaiTap_C3_Closing()	object sender, CancelEventArgs e		if(ltdl.IsConnOpen()) ltdl.DongKetNoi(); this.Close(); Application.Exit();	
4	cboBaitap_SelectedValueChanged()	object sender, EventArgs e		... DienDayDM(i+1); DienThongSo(i+1); rtbHuongdan.Text = "\t" + s + txtTuantu.Text; ...	
5	listView_Trinhdien_DoubleClick()	object sender, EventArgs e		image =new Bitmap(imageList_Trinhdien.Images[focusitem.ImageIndex],200,200); HienThiAnh hta= new HienThiAnh(); hta.pbxAnh.Image = image; hta.ShowDialog(this);	

6	btnVetruoc_Click()	object sender, System.EventArgs e		LamBaiTap_C2 fbt = new LamBaiTap_C2(); this.Hide(); fbt.ShowDialog();	
---	--------------------	--------------------------------------	--	---	--

### Danh sách các hàm thành phần

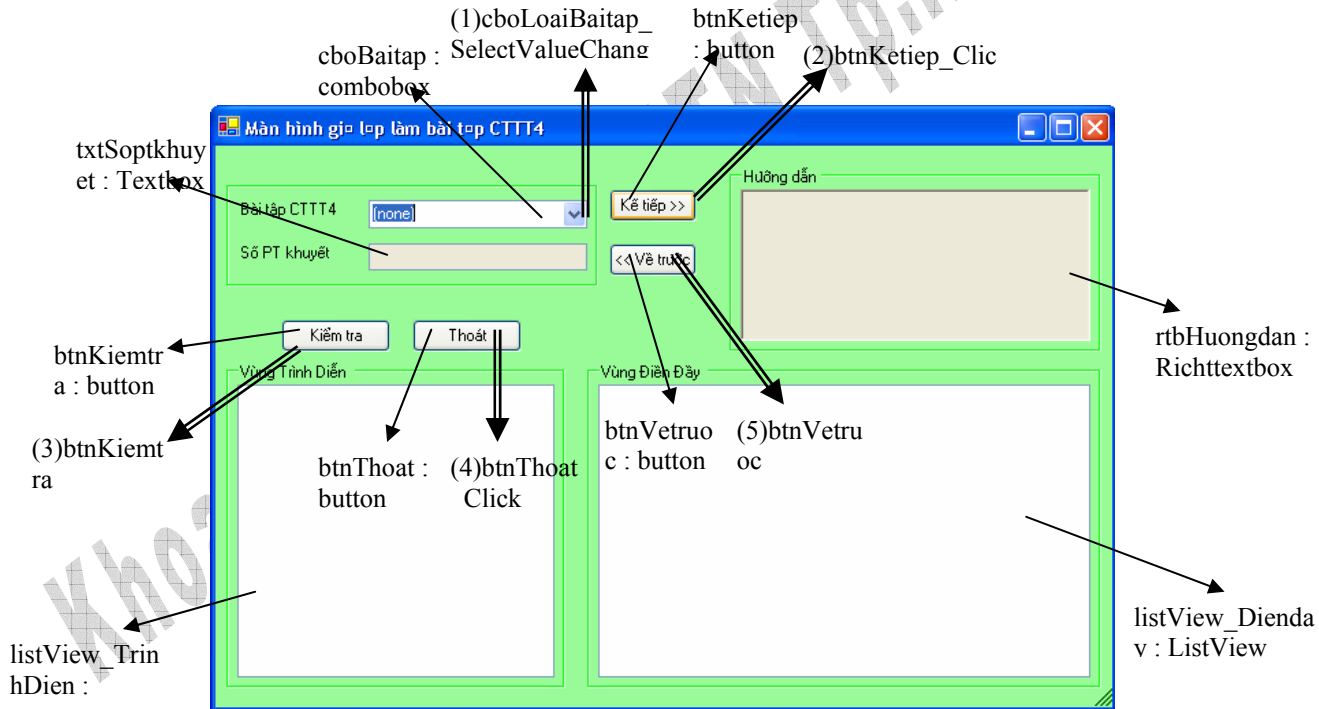
STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	LamBaiTap_C3()			ltdl = new LT_DuLieu(); xllv = new XuLy_ListView(); hd3 = new HuongDan_C3(); ltdl.MoKetNoi(); DienDayBaiTap(); m_nDoituongHientai=null; m_nDoituongTruoc=null;	Hàm khởi tạo
2	ReadBitmap2ByteArray()	Bitmap image	byte[]		Hàm chuyển đổi ảnh thành các byte
3	ReadByteArrayToBitmap()	byte[] content	Bitmap	MemoryStream stream = new MemoryStream(content); Bitmap image = new Bitmap(stream); return image;	Hàm chuyển đổi các byte thành ảnh
4	Mucdo()	Byte i	string	switch(i) { case 0: s="Dễ"; return s; case 1: s="Bình thường"; return s; case 2: s="Hỗn lộn"; return s; case 3: s="Rất hỗn lộn"; return s; }	Chuyển đổi mức độ ra chuỗi
5	Tuantu()	Byte i	string	if(i==0) return "Tăng"; return "Giảm";	chuyển đổi sự tuần tự ra chuỗi



6	DienThongSo()	int baitap_id		... dtheader = ltdl.LayThongSo_C3(baitap_id); ...	Lấy thông số của bài tập
7	TaoDanhSach()	ListViewXPDrापDrop lv,int SoPT		for(int i=0;i<SoPT;i++) { if(i<(SoPT/2)) xllv.Them1PT(lv,SoPT-i-1); else xllv.Them1PT(lv,i-SoPT/2); }	Tạo danh sách phần trình diễn theo mức hỗn lộn
8	Tong()	Bitmap bit	int	int s=0; byte[] b=ReadBitmap2ByteArray(bit); for(int i=0;i<b.Length;i++) s+=b[i]; return s;	Tính tổng các byte của hình
9	DienDayDM()	int baitap_id		... dtheader = ltdl.LayDanhMuc_C3(baitap_id); ... TaoDanhSach(listView_Trinhdien, imageList_Trinhdien.Images.Count);	lấy danh mục của bài tập
10	DienDayBaiTap()			... dtheader = ltdl.LayBaiTapCTTT3(); ... cboBaitap.Items.Add (dtheader.GetSqlString(1)); ...	lấy danh sách bài tập

**3.12/ Mô tả chi tiết lớp LAMBAITAP C4 :**

3.12.1/ Thiết kế giao diện :



3.12.2/ Thiết kế biến và xử lý :

**Danh sách các đối tượng thể hiện**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	listView_Trinhdien	ListviewXPDrpDrop		
2	listView_Dienday	ListviewXPDrpDrop		
3	btnKetiep	Button		
4	btnVetruoc	Button		
5	cboBaitap	ComboBox		
6	txtSPTK	TextBox		
7	btnKiemtra	Button		
8	btnThoat	Button		
9	rtbHuongdan	RichTextBox		
10				

**Danh sách các biến thành phần**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
-----	-----	----------	---------	---------

1	MaxElement=12	Const int	Biến hằng số phần tử trong tập
2	m_Image	Image	
3	xllv	XuLy_ListView	biến lớp xử lý listview
4	ArrK	int[]	mảng các phần tử index
5	ltdl	LT_DuLieu	

### Danh sách các sự kiện

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	btnKetiep_Click()	object sender, System.EventArgs e		LamBaiTap_C5 fbt = new LamBaiTap_C5(); this.Hide(); fbt.ShowDialog();	
2	btnVetruoc_Click	object sender, System.EventArgs e		LamBaiTap_C3 fbt = new LamBaiTap_C3(); this.Hide(); fbt.ShowDialog();	
3	btnKiemtra_Click	object sender, System.EventArgs e			
4	btnThoat_Click	object sender, System.EventArgs e		if(ltdl.IsConnOpen()) ltdl.DongKetNoi(); this.Close(); Application.Exit();	
5	LamBaiTap_C4_Closing()	object sender, CancelEventArgs e		if(ltdl.IsConnOpen()) ltdl.DongKetNoi(); this.Close(); Application.Exit();	
6	cboBaitap_SelectedValueChanged()	object sender, EventArgs e		... DienDayDM(i+1); DienThongSo(i+1); TaoMauTuanTu(i+1); rtbHuongdan.Text = s; ...	
7					

### Danh sách các hàm thành phần

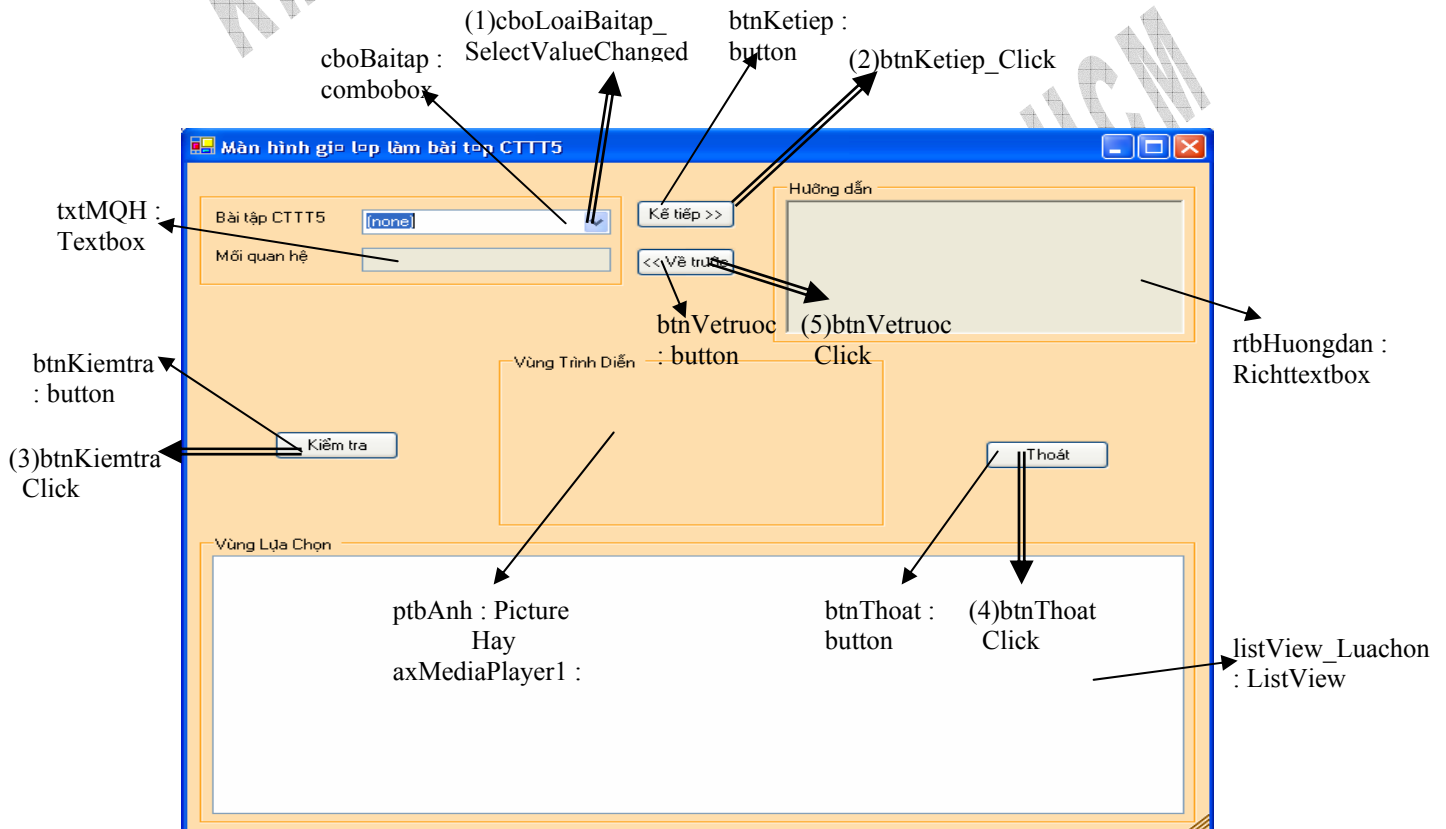
STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	LamBaiTap_C4()			ltdl = new LT_DuLieu(); ltdl.MoKetNoi(); DienDayBaiTap(); xllv = new XuLy_ListView();	Hàm khởi tạo
2	ReadByteArrayToBitmap()	byte[] content	Bitmap	MemoryStream stream = new MemoryStream(content); Bitmap image = new Bitmap(stream);	Hàm chuyển đổi các byte

				Return image;	thành ảnh
3	DienDayBaiTap()			<pre> ... dtreader = ltdl.LayBaiTapCTTT4(); ... cboBaitap.Items.Add (dtreader.GetSqlString(1)); ... </pre>	Lấy tất cả bài tập dạng này
4	DienDayDM()	int baitap_id		<pre> ... dtreader = ltdl.LayDanhMuc1_C4(baitap_id); ... xllv.Them1PT(listView_Trinhdien, imageList_Trinhdien.Images.Count-1); ... </pre>	Lấy danh mục nội dung của bài tập
5	DienThongSo()	int baitap_id		<pre> ... dtreader = ltdl.LayThongSo_C4(baitap_id); ... txtSPTK.Text = dtreader.GetInt32(0).ToString(); ... </pre>	Lấy các thông số của bài tập
6	TaoMauTuanTu()	int baitap_id		<pre> ... //vong lap tao mang //ngau nhien cac pt khuyet for(int i=1;i&lt;nk;i++) {     do     {         ArrK[i]=ran.Next(1,MaxElement- 1);     }while(ArrK[i]==ArrK[i-1]); } //dien day vao Vung dien day for(int i=1;i&lt;=MaxElement;i++) {     bDathem =false;     for(int j=0;j&lt;nk;j++)     {         if(i==ArrK[j])         {             imagelist_tam.Images.Add (imageList_Dienday.Images[0]);             bDathem = true;             break;         }     } } </pre>	Tạo mẫu tuần tự cho bài tập từ các đối tượng cơ bản

				<pre> if(!bDathem) { if(nPT&gt;0) { if(i%nPT!=0) imagelist_tam.Images.Add (imageList_Dienday.Images[i%nPT]); else imagelist_tam.Images.Add (imageList_Dienday.Images[nPT]); } } } xllv.ThemPTListView(listView_Dienday, imagelist_tam); ... </pre>	
--	--	--	--	--	--

**3.13/ Mô tả chi tiết lớp LAMBAITAP C5 :**

3.13.1/ Thiết kế giao diện :



## 3.13.2/ Thiết kế biến và xử lý :

**Danh sách các đối tượng thể hiện**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	listView_Luachon	ListViewXPDrappDrop		
2	picbox_Trinhdien	PictureBox		
3	btnKetiep	Button		
4	rtbHuongdan	RichTextBox		
5	cboBaitap	ComboBox		
6	txtMQH	TextBox		
7	btnKiemtra	Button		
8	btnVetruoc	Button		
9	btnThoat	Button		
10				

**Danh sách các biến thành phần**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	MAX=100	const int	biến hằng	
2	ltdl	LT_DuLieu	biến lớp lưu trữ dữ liệu	
3	hd5	HuongDan_C5	biến lớp hướng dẫn	
4	axMediaPlayer1	AxMediaPlayer	biến play âm thanh	
5	dsAmthanh	String[]	Danh sách âm thanh	
6	m_nMoiquanhe	Byte	mối quan hệ	
7	m_nTim	int	Chỉ số đang của phần tử đang hiển thị	
8	xllv	XuLy_ListView	biến lớp xử lý listview	

**Danh sách các sự kiện**

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	btnKetiep_Click()	object sender, System.EventArgs e		LamBaiTap_C6 fbt = new LamBaiTap_C6(); this.Hide(); this.timer1.Stop(); fbt.ShowDialog();	
2	btnVetruoc_Click	object sender, System.EventArgs e		LamBaiTap_C4 fbt = new LamBaiTap_C4(); timer1.Stop(); this.Hide(); fbt.ShowDialog();	
3	btnKiemtra_Click	object sender,			

		System.EventArgs			
4	btnThoat_Click	object sender, System.EventArgs		if(!tdl.IsConnOpen()) tdl.DongKetNoi(); timer1.Stop(); this.Close(); Application.Exit();	
5	LamBaiTap_C5_Closing()	object sender, CancelEventArgs		if(!tdl.IsConnOpen()) tdl.DongKetNoi(); timer1.Stop(); this.Close(); Application.Exit();	
6	cboBaitap_SelectedValueChanged()	object sender, EventArgs		... DienDayVungLuaChon(i+1); DienThongSo(i+1); rtbHuongdan.Text = "\t" + s1 + txtMQH.Text + s2; ... if(m_nMoiquanhe==0) { TaoDSAnhHienThi(i+1); ... } if(m_nMoiquanhe==1) { TaoDSAmThanhHienThi(i+1); ... } ... }	
7	listView_Luachon_Click()	object sender, EventArgs		Image Anhchon= imageList_Luachon.Images[item.ImageIndex] ; rtbHuongdan.AppendText("\n\t" + hd5.SoSanh2DT(Anhchon,imageList_Trinhdien.Images[m_nTim]));	
8	timer1_Tick()	object sender, EventArgs		... if(m_nMoiquanhe==0) { ... itmap b = new Bitmap(imageList_Trinhdien.Images[m_nTim],136, 88); picbox_Trinhdien.Image = b; ... } ... if(m_nMoiquanhe==1) {	

				<pre> ... axMediaPlayer1.FileName = dsAmthanh[m_nTim]; ... } ... </pre>	
--	--	--	--	---	--

### Danh sách các hàm thành phần

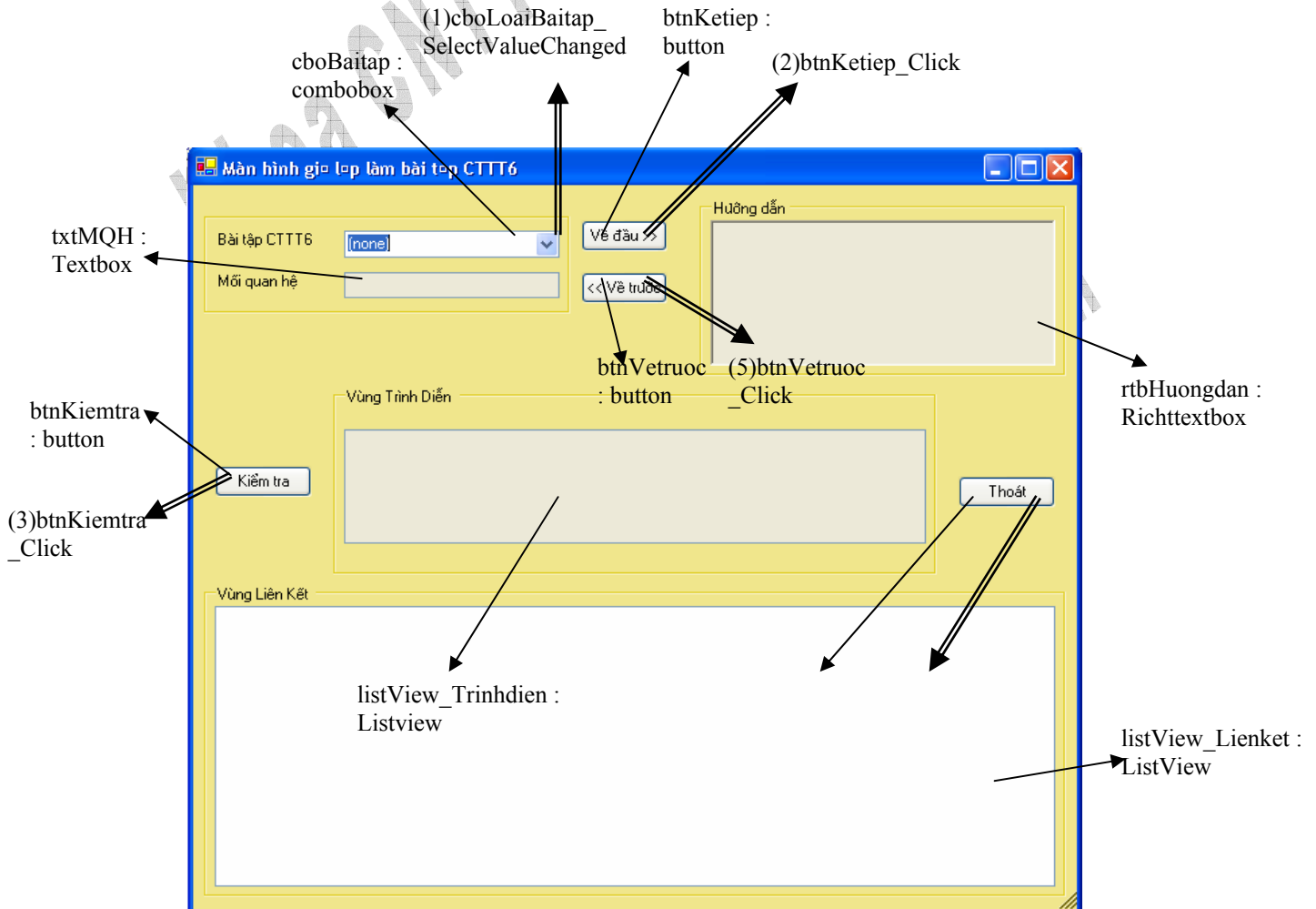
STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	LamBaiTap_C5()			<pre> ... ltdl = new LT_DuLieu(); ltdl.MoKetNoi(); DienDayBaiTap(); xllv = new XuLy_ListView(); hd5 = new HuongDan_C5(); </pre>	Hàm khởi tạo
2	ReadByteArrayToBitmap()	byte[] content	Bitmap	<pre> MemoryStream stream = new MemoryStream(content); Bitmap image = new Bitmap(stream); Return image; </pre>	Hàm chuyển đổi các byte thành ảnh
3	DienDayBaiTap()			<pre> ... dtreader = ltdl.LayBaiTapCTTT5(); ... cboBaitap.Items.Add (dtreader.GetSqlString(1)); ... </pre>	Lấy tất cả bài tập dạng này
4	DienDayVungLuaChon()	int baitap_id		<pre> ... dtreader = ltdl.LayDanhMuc1_C5(baitap_id); ... xllv.Them1PT(listView_Luachon, imageList_Luachon.Images.Count-1); ... </pre>	Lấy danh mục của bài tập
5	DienGiai()	Byte i	string	<pre> switch(i) {     case 0:         return "Hình ảnh và hình ảnh";     case 1:         return "Âm thanh và hình ảnh"; } return ""; </pre>	Diễn giải mối quan hệ
6	DienThongSo()	int baitap_id		<pre> dtreader = ltdl.LayThongSo_C5(baitap_id); ... m_nMoiquanhe = dtreader.GetByte(0); ... </pre>	Lấy thông số của bài tập
7	TaoDSAnhHienThi()	int baitap_id		...	Tạo danh



				dtreader = ltdl.LayDanhMuc2_C5(baitap_id); ...	sách ảnh sẽ hiển thị
8	TaoDSAmThanhHienThi()	int baitap_id		... dtreader = ltdl.LayDanhMuc3_C5(baitap_id); ... dsAmthanh[i]=dtreader.GetString(0); ...	Tạo danh sách âm thanh sẽ play

**3.14/ Mô tả chi tiết lớp LAMBAITAP\_C6 :**

3.14.1/ Thiết kế giao diện :



## 3.14.2/ Thiết kế biến và xử lý :

**Danh sách các đối tượng thể hiện**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	rtbHuongdan	RichTextBox		
2	listView_Trinhdien	ListViewXPDrpDrop		
3	listView_Lienket	ListViewXPDrpDrop		
4	btnVedau	Button		
5	btnVetruoc	Button		
6	btnThoat	Button		
7	cboBaitap	Combobox		
8	txtMQH	TextBox		
9	btnKiemtra	Button		
10	timer1	Timer		

**Danh sách các biến thành phần**

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	MAX=100	const int	hằng số	
2	m_nMoiquanhe	Byte	biến mối quan hệ	
3	m_nTim	int	biến chỉ số của danh sách phần tử đang hiển thị	
4	xllv	XuLy ListView	biến xử lý listview	
5	ltdl	LT_DuLieu	biến xử lý dữ liệu	
6				

**Danh sách các sự kiện**

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	btnVedau_Click()	object sender, System.EventArgs e		LamBaiTap_C1 fbt = new LamBaiTap_C1(); this.Hide(); this.timer1.Stop(); fbt.ShowDialog();	
2	btnVetruoc_Click	object sender, System.EventArgs e		LamBaiTap_C5 fbt = new LamBaiTap_C5(); this.Hide(); this.timer1.Stop(); fbt.ShowDialog();	
3	btnKiemtra_Click	object sender, System.EventArgs e			
4	btnThoat_Click	object sender,		if(ltdl.IsConnOpen())	

		System.EventArgs		<pre> ltdl.DongKetNoi(); timer1.Stop(); this.Close(); Application.Exit(); </pre>	
5	LamBaiTap_C6_Closing()	object sender, CancelEventArgs		<pre> if(ltdl.IsConnOpen())     ltdl.DongKetNoi(); timer1.Stop(); this.Close(); Application.Exit(); </pre>	
6	cboBaitap_SelectedValueChanged()	object sender, EventArgs		<pre> ... DienThongSo(i+1); DienDayVungLienKet(i+1); TaoDSTrinhDien(i+1); rtbHuongdan.Text = s1 + txtMQH.Text + s2; timer1.Start(); ... </pre>	
7	timer1_Tick()	object sender, EventArgs		<pre> ... HienThiVungTrinhDien(); m_nTim+=4; ... </pre>	

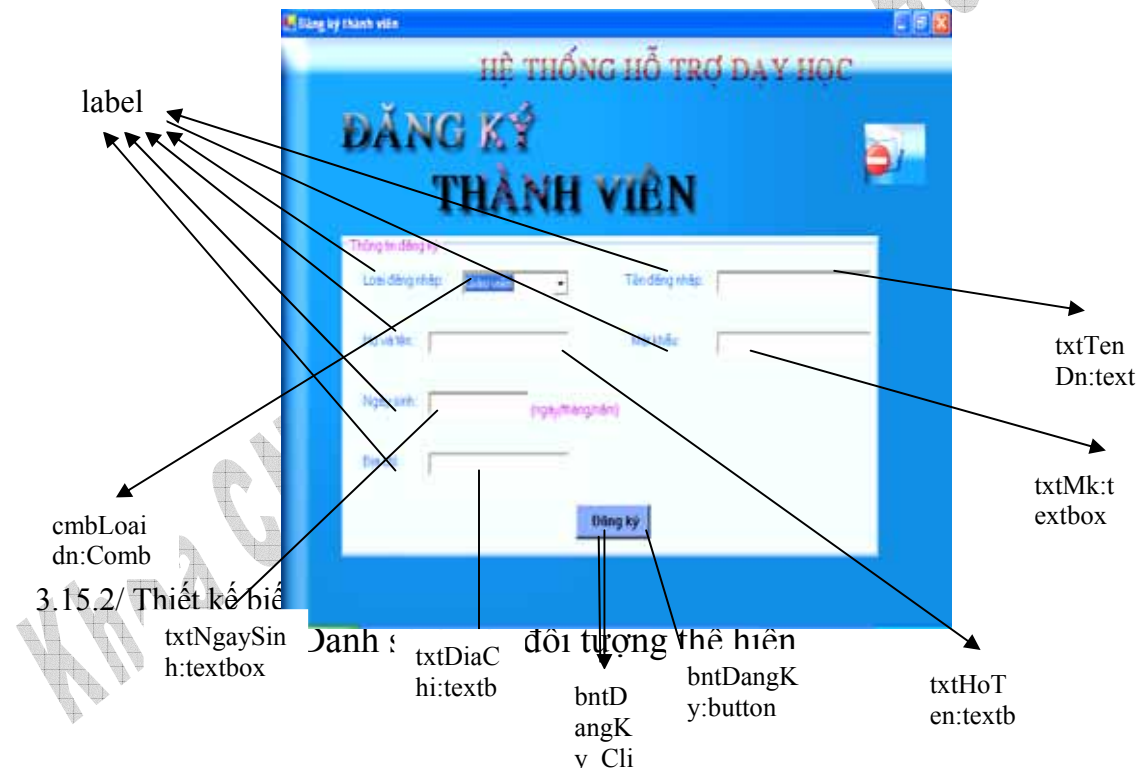
### Danh sách các hàm thành phần

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	LamBaiTap_C6()			<pre> ... ltdl = new LT_DuLieu(); ltdl.MoKetNoi(); DienDayBaiTap(); xllv = new XuLy_ListView(); </pre>	Hàm khởi tạo lớp
2	ReadByteArrayToBitmap()	byte[] content	Bitmap	<pre> MemoryStream stream = new MemoryStream(content); Bitmap image = new Bitmap(stream); Return image; </pre>	Hàm chuyển đổi các byte thành ảnh
3	DienDayBaiTap()			<pre> ... dtreader = ltdl.LayBaiTapCTTT6(); ... cboBaitap.Items.Add (dtreader.GetSqlString(1)); ... </pre>	Lấy tất cả bài tập của dạng này
4	DienDayVungLienKet()	int baitap_id		<pre> ... dtreader = ltdl.LayDanhMuc1_C6(baitap_id); ... xllv.Them1PT(listView_Lienket, imageList_Lienket.Images.Count-1); </pre>	Lấy tất cả phần tử và đưa vào vùng liên kết
5	DienGiai()	Byte i	string	<pre> switch(i) { </pre>	diễn giải mối quan

				<pre> case 0:     return "Hình ảnh"; case 1:     return "Câu hỏi trắc     nghiệm"; } return ""; </pre>	hệ ra chuỗi
6	DienThongSo()	int baitap_id		<pre> ... dtreader = ltdl.LayThongSo_C6(baitap_id); ... </pre>	Lấy tất cả thông số của bài tập
7	TaoDSTrinhDien()	int baitap_id		<pre> ... dtreader = ltdl.LayDanhMuc2_C6(baitap_id); ... </pre>	Lấy danh sách trình diễn từ CSDL
8	HienThiVungTrinhDien()			<pre> ... for(int i=m_nTim;i&lt;m_nCanTren;i++) { xllv.Them1PT(listView_Trinhdien,i); } ... </pre>	Hiển thị danh sách trình diễn
9					

**3.15/ Mô tả chi tiết lớp DangKy:**

3.15.1/ Thiết kế giao diện:



3.15.2/ Thiết kế biểu

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	cmbLoaidn	ComboBox	Loại thành viên đăng ký vào trong hệ thống	-Giáo viên -Học sinh - Chuyên gia
2	txtHoTen	Textbox	Tên của thành viên	
3	txtNgaySinh	Textbox		
4	txtDiaChi	Nt		
5	txtTendn	Nt		
6	txtMk	Nt		
7	bntDangKy	Button		

### Danh sách các biến thành phần

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	dl	LT DuLieu		

### Danh sách các sự kiện

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	bntDangKy_Click	<code>object sender, System.EventArgs e</code>		-nếu tên đăng nhập chưa tồn tại thì lưu -ngược lại: báo đã tồn tại rồi	

### Danh sách các hàm thành phần

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	KiemTraUsernameTruocKhiLuu	String username	Int	gọi hàm KiemTraUsername của lớp LT DuLieu	
2	KiemTraPswdTruocKhiLuu		Int	<code>if (txtMatKhai.Text.Length &gt;= 6) return 1; else return 0;</code>	

**3.16/ Mô tả chi tiết lớp DangNhap:**

3.16.1/ Thiết kế giao diện:



3.16.2/ Thiết kế biến và xử lý:

Danh sách các đối tượng thể hiện

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	cmbLoaidn	ComboBox	Loại thành viên đăng nhập trong hệ thống	-Giáo viên -Học sinh - Chuyên gia
2	txtTendn	Nt		
3	txtMk	Nt		
4	bntDangKy	Button		
5	bnrDangNhap	Button		

Danh sách các biến thành phần

STT	Tên	Kiểu/Lớp	Ý nghĩa	Ghi chú
1	dl	LT_DuLieu		

## Danh sách các sự kiện

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	bntDangKy_Click	<code>object sender, System.EventArgs e</code>		gọi form Đăng Ký	
2	bntDangNhap_Click	<code>object sender, System.EventArgs e</code>		-kiểm tra nếu đúng tên đăng nhập và mật khẩu thì cho đăng nhập - ngược lại thông báo lỗi	

## Danh sách các hàm thành phần

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	XacDinhUserandPwd	String username		Xác định loại thành viên và gọi hàm kiểm tra tương ứng	
2	KiemTraGiaoVien	String username			kiểm tra username đó có tồn tại hay chưa?
3	KiemTraHocSinh	String username			
4	KiemTraChuyenGia	String username			