





# ĐÈ TÀI

# PHÂN TÍCH THIẾT KẾ HỆ ĐA AGENT THEO PHƯƠNG PHÁP MaSE VÀ AgentTool







# Công nghệ phần mềm hướng Agent

# Mục lục

1. Mở đầu	3
2. Quy trình phát triển hệ thống phần mềm hướng Agent theo phương pháp luận l	
2.1 Khái quát các bước phát triển	
2.2 Pha phân tích	4
2.2.1 Bước 1: Xác định Goal (Capturing Goals)	4
2.2.2 Bước 2: Xác định Use Case (Applying Use Case)	6
2.2.3 Bước 3: Xây dựng Ontology (Building Ontology)	7
2.2.4 Bước 4: Hoàn thiện Role (Refining Roles)	10
2.3 Pha thiết kế	11
2.3.1 Bước 5: Xác định các lớp Agent (Creating Agent Classes)	11
2.3.2 Bước 6: Xây dựng các phiên hội thoại (Constructing Conversations)	12
2.3.3 Bước 7: Hoàn thiện Agent (Assembling Agent Classes)	
2.3.4 Bước 8: Thiết kế hệ thống (System Design):	14
3. AgentTool	14
3.1 Giới thiệu về AgentTool	
3.2 Thiết kế các hệ thống sử dụng agentTool	15
3.3 AgentTool hỗ trợ việc thiết kế bán tự động	15
3.4 Tính Di động	16
3.5 Các hình thức agentTool cơ bản	
4. Áp dụng phân tích và thiết kế hệ hỗ trợ dịnh vụ mua và bán điện thoại di động.	
4.1 Giới thiệu hệ hỗ trợ dịch vụ mua và bán máy điện thoại di động	
4.2 Phân tích hệ thống	18
4.2.1 Xác định Goal	18
4.2.2 Xác định Use case	
4.2.3 Xây dựng Ontology	
4.2.4 Hoàn thiện Role	
4.3 Thiết kế hệ thống	
4.3.1 Tạo lớp agent	
4.3.2 Xây dựng các conversation	
4.3.3 Hoàn thiện các agent	
4.3.4 Thiết kế hệ thống	
5. Đánh giá phương pháp luận MaSE	
5.1 Các khái niệm và các thuộc tính	
5.2 Các ký hiệu và các kỹ thuật mô hình hóa	
5.3 Quá trình phát triển	
5.4 Thực tế	
6. Kết luân	33

# Công nghệ phần mềm hướng Agent

<b>T</b>			~
Danh	muc	hinh	VĚ

Hình 1: Các bước phát triển hệ thống Multiagent theo phương pháp luận MaSE	4
Hình 2: Các bước xây dựng Ontology	8
Hình 3: Mô hình đối tượng MaSE hiện tại	
Hình 4: Mô hình đối tượng MaSE được mở rộng	
Hình 5: Biểu đồ phân cấp goal	
Hình 6: Biều đồ tuần tự của use case TimKiem	
Hình 7: Biều đồ tuần tự của use case ThuongLuong	
Hình 8: Biều đồ tuần tự của use case CapNhatThayDoi	
Hình 9: Biều đồ tuần tự của use case DatHang	22
Hình 10: Biều đồ tuần tự của use case HienThiKetQua	22
Hình 11: Mô hình role	
Hình 12: Biểu đồ task Thuong luong của role DaiLyPhanPhoi	26
Hình 13: Biểu đồ lớp agent	
Hình 14: Biểu đồ triển khai hệ thống	

# ĐỀ TÀI: PHÂN TÍCH THIẾT KẾ HỆ ĐA AGENT THEO PHƯƠNG PHÁP MaSE VÀ AgentTool

Giảng viên: *Ts. Nguyễn Mạnh Hùng* Học viên: *Đỗ Anh Tuấn* 

Lóp:CH10CNK2

**Tóm tắt:** Bài tiểu luận này cung cấp một tổng quan về bộ phương pháp luận MaSE và AgentTool để phân tích và thiết kế các hệ thống phần mềm hướng Agent được phát triển bởi nhóm nghiên cứu thuộc Viện Công nghệ Hàng Không Hoa kỳ (Air Force Intistute of Technology – AFIT). Nó sử dụng các sự trừu tượng được cung cấp bởi các hệ thống multiagent cho việc phát triển các hệ thống phần mềm thông minh và phân tán. Đồng thời qua tiểu luận này cũng trình bày việc áp dụng bộ phương pháp luận MaSE và AgentTool để phân tích và thiết kế hệ hỗ trợ dịch vụ mua và bán điện thoại di động

#### 1. Mở đầu

MaSE là một phương pháp luận được phát triển dựa trên cách tiếp cận hướng đối tượng và cung cấp một cách tiếp cận từ trên xuống (Top-Down). Quan điểm xây dựng của phương pháp luận này là xem Agent như mức trừu tượng cao hơn của một đối tượng: *mỗi Agent được xem là một đối tượng đặc biệt*. Khác với một đối tượng truyền thống trong đó các phương thức có thể được gọi bởi các đối tượng khác, các Agent tương tác với nhau thông qua hội thoại và hành động một cách tự chủ để hoàn thành mục đích của riêng mình cũng như mục đích chung của hệ thống. Ngoài ra, các Agent được xem như là một sự khái quát hóa đối tượng phù hợp với bài toán cụ thể, nó có thể có hoặc không có khả năng thông minh. Như DeLoach đã khẳng định, việc xem Agent như là một trừu tượng cao hơn của đối tượng khiến cho việc phân tích và thiết kế hướng Agent có thể thừa kế từ các phương pháp luận phát triển phần mềm hướng đối tượng.[6]

Quá trình phát triển hệ multiagent theo MaSE bao gồm có 2 pha: Pha phân tích và Pha thiết kế. Pha phân tích bao gồm các bước: Xác định Goal, Xác định các Use case, Xây dựng Ontology và Hoàn thiện Role. Pha thiết kế bao gồm các bước: Xác định các lớp Agent, Xây dựng hội thoại, Hoàn thiện Agent và Thiết kế hệ thống.

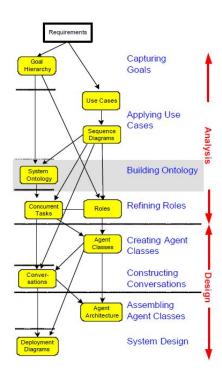
Toàn bộ quá trình phân tích thiết kế hệ thống theo phương pháp luận MaSE được hỗ trợ bởi công cụ AgentTool, bộ công cụ này hỗ trợ người thiết kế kiểm thử tương tác giữa các Agent và sinh mã tự động cho hệ thống.([1-4],[6])

Các phần tiếp theo của bài viết sẽ được cấu trúc như sau: Phần 2: trình bày quy trình phát triển hệ thống phần mềm hướng Agent theo phương pháp luận MaSE; Phần 3: giới thiệu về công cụ AgentTool; Phần 4: giới thiệu một áp dụng bộ phương pháp luận MaSE và AgentTool để phân tích và thiết kế hệ hỗ trợ mua và bán điện thoại di động; Phần 5: là sự một sự đánh giá về phương pháp luận MaSE; Phần 6: là kết luận.

# 2. Quy trình phát triển hệ thống phần mềm hướng Agent theo phương pháp luận MaSE

#### 2.1 Khái quát các bước phát triển

Quá trình phát triển hệ multiagent theo MaSE bao gồm có 2 pha: Pha phân tích (Analysis) và Pha thiết kế (Design). Pha phân tích bao gồm các bước: Xác định Goal (Capturing Goals), Xác định Use case (Applying Use Case), Xây dựng Ontology (Building Ontology) và Hoàn thiện Role (Refining Roles). Pha thiết kế bao gồm các bước: Xác định các lớp Agent (Creating Agent Classes), Xây dựng các phiên hội thoại (Constructing Conversations), Hoàn thiện Agent (Assembling Agent Classes) và Thiết kế hệ thống (System Design). Được thể hiện trong hình 1.



Hình 1: Các bước phát triển hệ thống Multiagent theo phương pháp luận MaSE

#### 2.2 Pha phân tích

#### 2.2.1 Bước 1: Xác định Goal (Capturing Goals)

Goal là một khái niệm để chỉ mục đích mà hệ thống cần đạt được. Mục đích của hệ thống ở đây được nhìn từ quan điểm của hệ thống nghĩa là các dịch vụ mà hệ thống có thể cung cấp. Goal sẽ được phân rã thành các Goal con, các Goal con này lại được tiếp tục phân rã và các Goal ở mức thấp hơn này sẽ không được coi là mục đích mà chỉ được xem xét để đưa vào các bước sau của pha phân tích.

Nhiệm vụ của bước này là chuyển toàn bộ đặc tả các yêu cầu hệ thống vào tập các Goal có cấu trúc. Như vậy có 2 bước trong việc xác định Goal: tập hợp các Goal và xây dựng cây phân cấp các Goal.

#### Tập hợp Goal

Bước này thực hiện các yêu cầu chức năng có trong tài liệu đặc tả hệ thống, mỗi yêu cầu chức năng mô tả bằng một Goal. Các yêu cầu chức năng được xác định bằng cách trả lời câu hỏi: "Hệ thống phải làm cái gì?" mà chưa cần quan tâm đến cách thực hiện nhiệm vụ đó như thế nào. Các Goal đầu tiên được xác định một cách trực quan thông qua việc xác định mục tiêu cần đạt được của hệ thống. Các Goal tiếp theo được xác định thông qua Goal trước bằng cách trả lời câu hỏi: "Muốn đạt được Goal X thì cần phải có cái gì?". Quá trình này được gọi là quá trình phân rã, các Goal được phân rã từ các Goal ban đầu sẽ trở thành các Goal con. Sự phân rã sẽ diễn ra với tất cả các Goal đã được phát hiện nhưng chưa được phân rã. Quá trình phân rã sẽ dừng lại khi các chức năng con sinh ra không phải là nhiệm vụ mức hệ thống, nghĩa là không thể đóng vai trò Goal của hệ thống. Các Goal không cần phân rã thêm có đặc điểm là khi cố gắng phân rã Goal này ta sẽ phải trả lời câu hỏi "Muốn hoàn thành việc này thì cần phải làm cái gì?", tức là tìm ra một cách thức thực hiện Goal chứ không phải là một Goal con.

# ■ Tổ chức cây phân cấp Goal

Bước này có nhiệm vụ tổ chức các Goal đã được xác định trong bước trước vào một sơ đồ phân cấp Goal (Goal Hierarchy Diagram). Một sơ đồ phân cấp Goal là một đồ thị có hướng và không có chu trình. Trong đó: các đỉnh biểu diễn các Goal, có tên trùng với tên của đích mà nó biểu diễn và các mũi tên chỉ ra quan hệ Goal cha – con và các quan hệ khác. Có hai trường hợp xảy ra: trường hợp thứ nhất: nếu đã xác định được Goal tổng thể của hệ thống thì đặt nó ở gốc của cây Goal, trường hợp thứ hai: nếu Goal tổng thể không xác định được trực tiếp từ yêu cầu thì phải kết hợp các Goal ở mức cao nhất thành một Goal tổng thể cho hệ thống. Các Goal còn lại có thể phân cấp thành các quan hệ cha – con hoặc ngang hang bằng cách lặp các thủ tục sau:

<u>Bước 1:</u> Các Goal được phân rã từ các Goal khác trong bước con trước phải là Goal con với Goal cha tương ứng

<u>Bước 2:</u> Nếu các Goal không được phân rã từ bất kỳ Goal nào (các Goal được xác định ngay ban đầu), để xác định quan hệ cha – con, thì trả lời câu hỏi "chúng có thể thực hiện một phần nhiệm vụ cho một Goal nào đó không?", nếu có nó sẽ trở thành Goal con mà nó hỗ trợ, nếu không phải xem xét lại Goal đó có cần thiết cho hệ thống hay không, nếu không cần thiết thì nó sẽ bị loại bỏ và ngược lại nếu cần thiết thì nó sẽ tạo thành một nhánh mới ngay từ nút gốc của cây Goal.

Trong cây phân cấp Goal có thể có 4 loại Goal sau:

Goal chung (Summary goal): là một Goal được tạo ra từ các Goal ngang hàng (thường là Goal tổng thể của hệ thống)

Goal phi chức năng (Non-functional goal): là các Goal không trực tiếp thực hiện một chức năng nào của hệ thống, nhưng là nhân tố kiểm tra tính đúng đắn của hệ thống. Các Goal này thường xuất hiện từ các yêu cầu phi chức năng của hệ thống chẳng hạn như độ tin cậy hay yêu cầu thời gian thực cho hệ thống.

Goal được kết hợp (Combined goal): là các Goal được tạo thành khi kết hợp hai hay nhiều Goal có chức năng giống nhau hoặc tương tự nhau.

Goal bị phân hoạch (Partitioned goal): là đích được phân hoạch hoàn toàn. Theo đó nếu tất cả các Goal con của nó được hoàn thành thì bản thân nó cũng được hoàn thành mà không cần thực hiện thêm nhiệm vụ nào nữa.

#### 2.2.2 Bước 2: Xác định Use Case (Applying Use Case)

Use case có thể hiểu là các mô tả về hành vi mà hệ thống cần thực hiện trong một trường hợp cụ thể. Các hành vi này được xuất phát từ mong muốn của người dùng.

Mục đích của bước này là tạo ra một tập các use case và các sơ đồ tuần tự (Sequence diagram) tương ứng nhằm hỗ trợ cho người phân tích hệ thống phát hiện được tập các role ban đầu và các đường truyền thông có thể có trong hệ thống.

Việc sử dụng use case trong MaSE được kế thừa từ phương pháp phân tích hướng đối tượng.

Có hai loại use case khác nhau dựa vào hoàn cảnh mà chúng mô tả:

<u>Use case chủ động:</u> mô tả hành vi của hệ thống trong trường hợp lý tưởng, nghĩa là các điều kiên đều thỏa mãn.

<u>Use case bị động:</u> mô tả hành vi mà hệ thống cần thực hiện trong trường hợp có lỗi, có ngoại lệ hoặc có sự cố nghiêm trọng.

Bước này cũng bao gồm hai bước con là tạo các use case và xây dựng biểu đồ tuần tư.

#### Tạo các use case

Các use case có thể được trích ra từ nhiều nguồn khác nhau: (i) đặc tả yêu cầu của hệ thống; (ii) Mong muốn của người dùng; (iii) Bản mẫu nhanh (nếu có). Về nguyên tắc mỗi một Goal được xác định trong bước 1 sẽ tương ứng với một use case, ngoại trừ các Goal bị phân hoạch (vì đối với các Goal này, khi hoàn thành các use case của các Goal con thì bản thân nó cũng được hoàn thành).

Trong các use case tương ứng với mỗi Goal cha, dãy hành động thuộc use case của Goal con sẽ được coi là một Goal đơn, nghĩa là nó được xem tương ứng với một hành động đơn bình thường khác và cũng được biểu diễn bằng một sự kiện đầu vào và một hành động được thực hiện.

Các use case ứng với các Goal thuộc lá của cây Goal sẽ được trích dẫn trước, sau đó ngược dẫn lên phía gốc của cây Goal và dừng lại khi gặp một Goal bị phân hoạch. Goal bị phân hoạch sẽ không cần use case bởi vì các use case tương ứng của nó chính là phép hợp đơn giản của các use case tương ứng với các đích con của nó mà không cần bổ xung thêm bất kỳ sự kiện hay hành động nào.

# Xây dựng biểu đổ tuần tự

Một biểu đồ tuần tự là một dãy các sự kiện diễn ra giữa các role đã được xác định trong các use case. Nó được xây dựng dựa trên các role và quan hệ tương tác giữa các role này với nhau.

Một biểu đồ tuần tự bao gồm:

 Các role liên quan đến use case cần biểu diễn, các role này được đặt phía trên của biểu đồ

- Các đường nối từ các role thẳng xuống dưới là các đường biểu thị cho thời gian hoạt động của các role
- Các mũi tên nối từ role này đến role kia biểu thị một tương tác giữa hai role, theo chiều mũi tên. Nhãn kèm theo mũi tên sẽ biểu thị tên của sự kiện tương tác đó.
- Tuần tự các sự kiện sảy ra trong use case sẽ là tuần tự các mũi tên đi từ trên xuống dưới

Việc chuyển từ một use case sang một biểu đồ tuần tự có thể tiến hành trực tiếp như sau: Mỗi use case có thể tương ứng với một biểu đồ tuần tự. Tuy nhiên trong một số trường hợp mà use case quá phức tạp, nó có thể cần đến nhiều hơn một biểu đồ tuần tự để mô tả hoạt động. Trong trường hợp này, cũng có thể tách use case thành các use case nhỏ hơn và đơn giản hơn, mỗi use case chỉ cần một biểu đồ dãy như trường hợp lý tưởng ban đầu.

Việc xây dựng biểu đồ tuần tự được bắt đầu bằng việc xác định các role cần thiết phải tham gia vào biểu đồ. Role có thể hiểu là một tập các nhiệm vụ có liên quan chặt chẽ đến nhau mà các Agent trong hệ thống cần đảm nhiệm để đạt tới đích. Mỗi role có thể thực hiện một hay nhiều Goal của hệ thống.

Để xác định các role, người phát triển có thể sử dụng kỹ thuật trích danh từ. Từ các use case đã có trong bước trước, người phát triển sẽ tiến hành duyệt và tìm ra các danh từ chỉ các đối tượng có chức năng cụ thể nào đó trong use case trước đó. Tiếp theo, người phát triển sẽ xem xét lại các danh mục các danh từ này và loại đi các danh từ chỉ cùng một đối tượng. Các danh từ còn lại sẽ là các role sẽ có mặt trong sơ đồ tuần tự tương ứng với use case đó. Sau khi đã có các role thì trong bước tiếp theo người phát triển sẽ tiến hành biểu diễn tuần tự các sự kiện của sơ đồ tuần tự trong AgentTool.

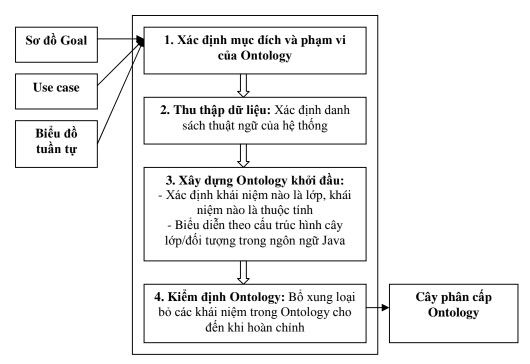
#### 2.2.3 Bước 3: Xây dựng Ontology (Building Ontology)

Ontology có thể xem là một tập các khái niệm để biểu diễn thông tin và tri thức trong trao đổi giữa các Agent hay nói cách khác, Ontology là một hệ tri thức chung giữa các agent, bao gồm các khái niệm được dung trong một miền tri thức nhất định. Theo quan điểm MaSE, ontology là tập hợp các khái niệm có thể sử dụng như là các tham số chứa trong các thông điệp trao đổi giữa các Agent.

Trong MaSE xây dựng Ontology gồm 4 bước chính: Xác định mục đích và phạm vi của ontology; Thu thập dữ liệu; Xây dựng Ontology khởi đầu; Hoàn thiện Ontology.

Kết quả của quá trình này là một sơ đồ phân cấp các khái niệm có thể đáp ứng yêu cầu hoạt động của hệ thống.

Các bước tiến hành xây dựng Ontology được thể hiện trong hình 2.



Hình 2: Các bước xây dựng Ontology

<u>Bước 1: Xác định mục đích và phạm vi của Ontology:</u> Bước này xác định lĩnh vực mà Ontology cần biểu diễn. Nghĩa là người phân tích phải trả lời câu hỏi:"*Tại sao Ontology được xây dựng và phạm vi ý định sử dụng Ontology của hệ thống*". Khi một hệ thống kế thừa Ontology từ một hệ thống khác, Ontology được kế thừa đó phải được mô tả chi tiết và chính xác phạm vi biểu diễn của nó. Từ đó, người phân tích so sánh phạm vi này với phạm vi lĩnh vực của hệ thống mình đang xây dựng: Nếu phạm vi của mình rộng hơn thì phải bổ xung thêm các khái niệm; Nếu phạm vi hẹp hơn thì phải loại bỏ các khái niệm không cần thiết ra khỏi Ontology được kế thừa.

Để xác định phạm vi của Ontology cần thực hiện các bước sau:

- Xem xét toàn bộ use case đã được mô tả trong bước xác định use case và sơ đồ Goal của hệ thống trong bước xác định Goal
- Xác định toàn bộ các thông tin và kiểu dữ liệu mà hệ thống sẽ dùng, đồng thời xác định được mức độ chi tiết cần mô tả của các kiểu dữ liệu này.

Người phân tích có thể sử dụng kỹ thuật khoanh vùng và thu hẹp dần các miền tri thức để xác định phạm vi của Ontology. Để xác định *mức độ chi tiết của các tri thức và khái niệm*, người phân tích tiếp tục sử dụng các kỹ thuật:

- Phân rã miền tri thức ban đầu thành các miền con cho đến khi không thể phân rã thêm được nữa.
- Khoanh vùng các miền tri thức có liên quan đến bài toán và loại bỏ các vùng không liên quan.

<u>Bước 2: Thu thập dữ liệu:</u> Sau khi xác định được các loại dữ liệu và mức độ chi tiết của chúng, người thiết kế phải lập ra một danh sách liệt kê tất cả các danh từ có mặt trong cây phân cấp Goal (đã có ở bước xác định Goal) và các use case. Danh sách này được gọi là "danh sách thuật ngữ" của hệ thống. Từ tập thuật ngữ này, người thiết kế phải chọn ra các danh từ có thể biểu diễn một kiểu dữ liệu cần thiết cho hệ thống bằng

cách xem xét tất cả các kiểu dữ liệu cần truyền đi trong các use case. Bước này thực chất là tìm ra các khái niệm có trong các message và các khái niệm ở mức nhỏ hơn để các agent có thể hiểu được các message mà nó nhận được. Sau bước này ta nhận được các khái niệm cần có trong ontology.

Bước 3: Xây dựng Ontology khởi đầu: Để xây dựng được Ontology khởi đầu, người thiết kế phải chuyển toàn bộ các khái niệm đã thu được trong bước 2 vào tập các lớp và các thuộc tính của chúng. Vấn đề chính là làm sao xác định được khái niệm nào là lớp, khái niệm nào là thuộc tính, điều này phụ thuộc vào việc xác định mức độ chi tiết của mỗi khái niệm. Thông thường các khái niệm biểu diễn mức độ chi tiết thấp nhất thì nên lấy làm thuộc tính, các khái niệm là tổ hợp của nhiều khái niệm con thì nên xem là lớp. Tuy nhiên, không phải mỗi khái niệm đều phải mô tả đầy đủ các thuộc tính như ý nghĩa thực tế của nó. Trong Ontology, chỉ các thuộc tính cần thiết để thực hiện các đích con và đích tổng thể của hệ thống mới được đưa vào làm thuộc tính cho các đối tượng tương ứng. Để xây dựng được ontology khởi đầu, người phân tích có thể lựa chọn một trong hai cách:

- Kế thừa từ một Ontology của một hệ thống khác: Người phân tích phải xác định rõ phạm vi biểu diễn của Ontology của hệ thống được kế thừa và phạm vi tri thức của hệ thống mình đang xây dựng. Nếu phạm vi tri thức của hệ thống được kế thừa rộng hơn, thì loại bỏ các thông tin dư thừa và bổ xung các thông tin chi tiết hơn. Nếu phạm vi tri thức của hệ thống được kế thừa nhỏ hơn thì phải bổ xung các tri thức về miền lĩnh vực mới.
- Xây dựng Ontology ngay từ đầu: người phân tích phải chuyển toàn bộ các khái niệm đã thu được trong bước 2 vào tập các lớp và các thuộc tính của chúng. Thông thường, các khái niệm biểu diễn mức độ chi tiết thấp nhất thì nên lấy làm thuộc tính, các khái niệm là tổ hợp của các khái niệm con thì nên tạo thành lớp.

<u>Bước 4: Kiếm định Ontology:</u> Trong bước này, người phân tích phải chứng thực được rằng Ontology vừa xây dựng đã thỏa mãn yêu cầu của hệ thống bằng cách xem xét lại toàn bộ các tình huống đã được mô tả trong các use case và sơ đồ tuần tự:

- Nếu phát hiện thấy một thiếu sót nào đó thì khái niệm tương ứng phải được bổ xung ngay vào Ontology;
- Nếu phát hiện thấy có khái niệm nào không cần thiết thì phải loại bỏ ngay ra khỏi Ontology

Quá trình này được lặp lại cho đến pha cài đặt hệ thống hoặc cho đến khi người thiết kế thấy rằng, Ontology đã hoàn toàn thỏa mãn yêu cầu hoạt động của hệ thống. Kết quả của bước này là một cây phân cấp Ontology hoàn chỉnh, có thể đáp ứng được yêu cầu sử dụng trong các bước tiếp theo của quá trình phân tích thiết kế và đảm bảo đầy đủ các khái niệm cần cho tương tác của hệ đa agent.

Những bước xây dựng Ontology theo phương pháp luận MaSE là khá rõ ràng và tách biệt. Tuy nhiên, trên thực tế các bước này đều nằm trong vòng đời phát triển Ontology nên có thể được thực hiện lồng vào nhau. Mặt khác, do đặt bước xây dựng Ontology trong vòng đời phát triển chung của hệ thống nên nếu có thay đổi trong các bước đầu tiên như xác định Goal, xác định cây phân cấp Goal, xác định use case và các biểu đồ tuần tự thì toàn bộ quá trình xây dựng Ontology cũng phải thay đổi theo. Và

ngược lại, nêu cây phân cấp Ontology thay đổi thì các bước sau của phân tích thiết kế cũng cần thay đổi cho phù hợp

#### 2.2.4 Bước 4: Hoàn thiện Role (Refining Roles)

Role là khái niệm dùng để chỉ các thành viên (đối tượng) thực hiện một (hoặc một số) vai trò nhất định trong hệ thống. Mục tiêu của bước này là chuyển tập các Goal của hệ thống vào tập các role cùng với các nhiệm vụ của nó.

Thông thường, việc chuyển đổi từ Goal sang Role là tương ứng 1-1, nghĩa là mỗi Role thực hiện một Goal. Tuy nhiên, trong một số trường hợp, một Role có thể tương ứng với nhiều Goal khi các Goal này có quan hệ chặt chẽ hoặc gần giống nhau. Trong bước này, người phát triển hệ thống sẽ phải xem xét lại các role đã có trong trong bước xây dựng sơ đồ tuần tự, thêm bớt, sửa đổi cho phù hợp.

Các giao tiếp với bên ngoài sẽ được xây dựng thành một role riêng biệt và hoạt động tương tự như một tương tác từ một nguồn tài nguyên bên ngoài với phần còn lại của hệ thống. Các thành phần được coi là nguồn tài nguyên bên ngoài bao gồm: cơ sở dữ liệu, các file, hệ thống bổ xung và con người. Sau khi xác định được các role phù hợp của hệ thống, phải xác định được tập các giao tiếp ban đầu giữa các role này. Các tương tác này được trích từ các use case của bước xác định use case.

Nếu giữa các role trong biểu đồ tuần tự có một hoặc một số sự kiện liên quan đến nhau thì giữa các role tương ứng trong biểu đồ này sẽ có một giao thức tương tác, bên khởi đầu trùng với bên khởi đầu các sự kiện trong use case. Sơ đồ Role ban đầu này sẽ được chi tiết hóa bằng cách gán các task cho các role. Task là một nhiệm vụ cụ thể, chi tiết mà các role phải thực hiện nhằm đạt được đích mà nó có trách nhiệm phải thực hiên.

Thông thường, mỗi Goal nên cho tương ứng với một task, vì muốn đạt được một Goal thì ít nhất một task cần phải hoàn thành. Trong trương hợp đặc biệt, đối với các Goal phức tạp thì có thể cần đến hơn một task để giải quyết nó. Nên tách Goal đó thành các Goal bé hơn, đơn giản hơn sao cho một task đã có thể giải quyết được. Dựa vào các Goal mà mỗi role phải thực hiện, các task được tạo ra bằng cách trả lời câu hỏi:"*Role sẽ thực hiện các Goal đó như thế nào?*" Sau khi các task được gán vào các role xác định, các giao thức giữa các role trong sơ đồ role ban đầu sẽ được xác định cụ thể bởi các task liên quan.

Sơ đồ bên trong task: Sau khi xác định được các role, các task sẽ được gán cho mỗi role để mô tả cách mỗi role cư xử và hoạt động nhằm đạt được đích mà nó hướng tới. Các role có thể hoạt động với hai loại tương tác:

- Tương tác trong: các tương tác giữa các task trong cùng một role
- Tương tác ngoài: các tương tác giữa các task của các role khác nhau

Trong sơ đồ chi tiết, các tương tác trong được biểu diễn bằng các mũi tên nét đứt, còn các tương tác ngoài được biểu diễn bằng các mũi tên nét liền. Các tương tác này sẽ định hướng cho việc thiết kế các cuộc hội thoại (conversation) trong pha thiết kế sau này. Có hai loại task khác nhau:

- Task chủ động: là task tự động bắt đầu khi role được sinh ra. Task này không cần điều kiện kích hoạt ban đầu, nó tự động đi đến trạng thái ban đầu và thực

- hiện các hành động cho đến khi gặp trạng thái kết thúc hoặc role tương ứng của nó bị hủy bỏ.
- Task bị động: là task chỉ được thực hiện khi có sự kiện kích hoạt từ bên ngoài (thường là một thông điệp từ task khác). Task này hoạt động cho đến khi gặp trang thái kết thúc hoặc có một task khác yêu cầu nó kết thúc.

Hoạt động bên trong của các task được mô tả theo sơ đồ máy trạng thái hữu hạn (finite state machine), bao gồm các trạng thái và các chuyển tiếp giữa các trạng thái. Tại các trạng thái, các task được thực hiện thông qua các hàm cụ thể hoặc chờ cho tới khi một sự kiện nào đó xảy ra. Cú pháp của các trạng thái chuyển tiếp là tuy thuộc vào kiểu tương tác mà các chuyển tiếp đó tham gia là tương tác trong hay tương tác ngoài. Các chuyển tiếp tham gia vào tương tác được mô tả theo cú pháp: <trigger><[guard]><transmissions> . Nghĩa là: nếu có sự kiện trigger xảy ra mà nó đang giữ điều kiện guard thì nó sẽ gửi đi các thông điệp transmissions và chuyển sang trạng thái tiếp theo. Trong các sơ đồ task, có một số trường hợp các điều kiện chuyển tiếp đồng loạt được thỏa mãn, nghĩa là các chuyển tiếp có thể diễn ra đồng thời thì thứ tự ưu tiên các chuyển tiếp cần thực hiện được xác định như sau:

- Chuyển tiếp có điều kiện [guard] là một loại sự kiện thuộc loại tương tác trong;
- Chuyển tiếp có *transmission* là một sự kiện thuộc loại tương tác trong
- Chuyển tiếp có receive() từ các role khác (tương tác ngoài)
- Chuyển tiếp có *send()* gửi thông điệp đến role khác;
- Chuyển tiếp chỉ có điều kiện [guard]

Như vậy ưu tiên trước hết được xác định dành cho các chuyển tiếp có trao đổi thông tin với các task khác, mức thứ 2 là các tương tác thuộc loại tương tác trong, mức thứ ba là các tương tác có nhận thông điệp được ưu tiên hơn tương tác gửi thông điệp.

Kết quả của pha phân tích là một tập các role và các task tương ứng nhằm hoàn thành mục đích của hệ thống. Hoạt động bên trong và các tương tác bên ngoài giữa các task là nhân tố quan trong cho pha thiết kế sau này.

#### 2.3 Pha thiết kế

# 2.3.1 Bước 5: Xác định các lớp Agent (Creating Agent Classes)

Trong bước này các lớp Agent sẽ được tạo ra từ các role đã được xác định trong bước 4 của pha phân tích, bao gồm: Phân chia các role cho các lớp Agent; Xác định các phiên hội thoại xuất hiện giữa các lớp agent.

Kết quả cần thu được của bước này sơ đồ các lớp agent (agent classes diagram). Sơ đồ này mô tả hệ thống một cách tổng thể theo các lớp agent và các phiên hội thoại liên lạc giữa chúng.

Để đảm bảo các Goal hệ thống đều được thực hiện trong pha thiết kế, mỗi role phải được đảm nhiệm bởi ít nhất một lớp agent, đôi khi cũng có các role được đảm nhiệm bởi nhiều lớp agent. Trong trường hợp một lớp agent đảm nhiệm nhiều role thì các agent của lớp đó sẽ luân phiên đảm nhiệm công việc của từng role. Trường hợp này xảy ra khi các role phục vụ cho các Goal liên quan chặt chẽ với nhau hoặc có khối lượng tương tác với nhau lớn, chúng được xếp vào cùng một lớp agent nhằm giảm chi

phí truyền thông. Người thiết kế có thể dễ dàng thay đổi các role giữa các lớp agent, điều này cho phép họ điều chỉnh hệ thống theo mục đích của mình như sau:

- Nếu giưa 2 role có lưu lượng trao đổi thông tin lớn thì nên xếp chúng vào cùng một lớp agent để giảm chi phí truyền thông.
- Nếu cả hai role có khối lượng tính toán lớn và phức tạp thì nên xếp chúng vào hai lớp agent khác nhau để giảm khối lượng tính toán đồng thời tận dụng được khả năng xử lý đồng thời giữa các agent.

Nhiệm vụ còn lại của bước này là xác định các phiên hội thoại xuất hiện giữa các lớp agent để hoàn thiện sơ đồ lớp agent của hệ thống. Các phiên hội thoại được xác định từ các quan hệ giữa các role mà các lớp agent tương ứng cần thực hiện. Chẳng hạn, hai lớp agent A và B lần lượt thực hiện các role 1 và 2 tương ứng. Nếu giữa role 1 và 2 có một giao thức được xác định trong sơ đồ role thì giữa hai lớp agent A và B cũng xuất hiện một phiên hội thoại theo chiều tương ứng.

Các phiên hội thoại biểu diễn tương tác giữa các lớp agent diễn ra ở mức ngữ nghĩa thông qua ontology. Dựa vào Ontology đã được xây dựng trong bước 3, các lớp agent có cách biểu diễn tri thức khác nhau sẽ trao đổi và hiểu được lẫn nhau dựa trên tri thức chung trong Ontology.

## 2.3.2 Bước 6: Xây dựng các phiên hội thoại (Constructing Conversations)

Một phiên hội thoại biểu diễn một phiên liên lạc giữa hai agent, nghĩa là các agent sẽ tiến hành gửi đi các yêu cầu và đáp ứng lại các yêu cầu từ phía agent kia. Trong MaSE, mọi liên lạc giữa hai agent bất kỳ đều phải thông qua các phiên hội thoại được hoạt động theo cơ chế socket và cổng (port), với các chuẩn của giao thức TCP/IP.

Nhiệm vụ của bước này là thiết kế chi tiết kiến trúc bên trong của các phiên hội thoại đã được xác định trong bước 5. Đối với mỗi phiên hội thoại phải làm sáng tỏ được cách thức và cơ chế hoạt động của mỗi bên agent tham gia vào phiên hội thoại đó

Mỗi phiên hội thoại bao gồm hai sơ đồ lớp truyền thông (Communication class diagram), một cho bên khởi xướng và một cho bên đáp ứng phiên hội thoại. Mỗi sơ đồ lớp truyền thông được biểu diễn như một sơ đồ máy trạng thái hữu hạn tương tự như các task đồng thời trong Bước 4.

Performative là một khái niệm liên quan đến các thông điệp được trao đổi trong các phiên hội thoai. Nó là một tập các từ thuộc thể loại mệnh lệnh thức (lời nói mang ý nghĩa hành động), được quy định cho mỗi hệ thống.

Trong bước 5, các phiên hội thoại đã được xác định từ các tương tác giữa các role của các lớp agent khác nhau. Do vậy các sơ đồ lớp truyền thông cũng được xác định tương ứng với các task mà các role đó thực hiện. Thông thường, mỗi sơ đồ task sẽ tương ứng với một sơ đồ phiên hội thoại cho bên tham gia tương ứng.

Tại các trạng thái của phiên hội thoại, người thiết kế phải xác định chính xác tên các hàm, các biến và các tham số cho các hàm mà không được để một cách khái quát như các hàm và thủ tục trong các trạng thái của task.

Tại các chuyển tiếp của sơ đồ phiên hội thoại, người thiết kế phải chi tiết hóa dựa trên các chuyển tiếp trong sơ đồ task tương ứng. Cùng dựa trên cú pháp của chuyển tiếp receive (message)><[guard]><send(message)>< nhưng được cụ thể hóa nội</pre>

dung các thông điệp (message) dựa vào việc sử dụng Ontology của hệ thống đã được thiết kế trong bước 3.

Sau khi các thông tin từ sơ đồ task được ánh xạ vào như một phần của phiên hội thoại, người thiết kế phải kiểm tra lại để đảm bảo các điều kiện khả thi cho phiên hội thoại: mỗi message gửi đi phải có ít nhất một bên nhận và xử lý, không có vòng lặp vô hạn trong các sơ đồ trạng thái. Mỗi phiên hội thoại được coi là hợp lệ nếu nó thỏa mãn các điều kiện sau:

- Trong mỗi sơ đồ của các bên không có vòng lặp vô hạn (deadlock), nghĩa là nếu trong sơ đồ có xuất hiện chu trình thì chu trình này phải có điều kiện dừng và điều kiện dừng này là có khả năng đạt được.
- Với mỗi lớp agent liên quan đến phiên hội thoại, tại mỗi thời điểm nó chỉ được ở trong một trạng thái xác định
- Các trạng thái đều có thể sử dụng trong một điều kiện nhất định (không có trạng thái không bao giờ được sử dụng)
- Mỗi thông điệp được gửi đi thì bên đối diện phải có khả năng nhận được thông điệp đó.

Việc kiểm tra các điều kiện này có thể tiến hành một cách thủ công hoặc bán tự đông với sư trơ giúp của agentTool.

Trong việc thiết kế các phiên hội thoại, người thiết kế phải đối mặt với vấn đề là nên lựa chọn theo hướng ít phiên hội thoại dạng phức tạp hoặc nhiều phiên hội thoại dạng đơn giản. Việc sử dụng các phiên hội thoại đơn giản có ưu điểm là dễ theo dõi, quản lý và đồng thời tiết kiệm được tài nguyên mạng như kênh truyền, cổng, kết nối...Lý do là khi kết thúc, các tài nguyên này được giải phóng và khi cần thiết, chúng mới bị chiếm dụng trở lại. Nhưng nhược điểm của cách tiếp cận này là làm chậm tốc độ do phải tốn nhiều thời gian cho việc thiết lập và giải phóng các kết nối truyền thông. Kiểu các phiên hội thoại phức tạp cho kết quả ngược lại, tăng tốc độ nhưng tốn tài nguyên do nó vẫn giữ đường truyền ngay cả khi không truyền thông điệp mà đang tính toán tại các trạng thái.

#### 2.3.3 Bước 7: Hoàn thiện Agent (Assembling Agent Classes)

Bước này nhằm thiết kế chi tiết các tương tác bên trong (self interaction) của mỗi lớp agent trong hệ thống. Bao gồm hai bước con: thiết kế kiến trúc bên trong agent và thiết kế các thành phần trong kiến trúc đó.

Thiết kế kiến trúc: Để xác định kiến trúc bên trong của các agent, người thiết kế có thể tự thiết kế kiến trúc của mình hoặc có thể chọn một kiến trúc sẵn có như BDI. Trong trường hợp không sử dụng kiến trúc sẵn có, người thiết kế có thể xây dựng các thành phần từ các nhiệm vụ của các role mà lớp agent tương ứng đảm nhiệm. Sau khi xác định được các thành phần trong kiến trúc của mỗi agent, nhiệm vụ tiếp theo là thiết kế quan hệ giữa các thành phần này sao cho phù hợp với sơ đồ role của hệ thống đã được phân tích trong bước hoàn thiện role. Để đảm bảo quan hệ giữa các thành phần thống nhất với quan hệ giữa các nhiệm vụ trong sơ đồ role, các quan hệ này sẽ được thiết kế dựa trên các tương tác giữa các nhiệm vụ trong sơ đồ role:

 Các tương tác giữa các nhiệm vụ sẽ trở thành một quan hệ trong kiến trúc này và được biểu diễn bằng một mũi tên nét liền, chiều của mũi tên trùng với chiều của tương tác tương ứng trong sơ đồ role (Một số tương tác bên ngoài cũng có thể trở thành các quan hệ bên trong nếu các role của các nhiệm vụ tương ứng được thực hiện bởi một lớp agent)

- Các tương tác bên ngoài giữa các role được đảm nhiệm bởi các lớp agent khác nhau sẽ trở thành các quan hê bên ngoài.

Thiết kế các thành phần trong kiến trúc: Nhiệm vụ của bước con này là thiết kế chi tiết các thành phần của kiến trúc đã được thiết kế ở trên. Việc này được hoàn thành bằng việc gán các thuộc tính và các hàm (thủ tục) chính cho mỗi thành phần. Để xác định được các hàm và thuộc tính, người thiết kế phải duyệt trong tất cả các trạng thái của các phiên hội thoại mà thành phần tương ứng tham gia như trong bước 6 (Xây dựng hội thoại) đã xác định cụ thể tên các hàm cần thiết.

Trước khi kết thúc bước thiết kế kiến trúc bên trong của agent, người thiết kế phải hoàn thành việc thiết kế Ontology riêng cho mỗi thành phần của mỗi lớp agent nếu điều đó là cần thiết. Mỗi lớp agent được thiết kế thêm phần Ontology riêng nếu như Ontology chung của hệ thống không giúp nó biểu diễn đầy đủ được tri thức của mình.

# 2.3.4 Bước 8: Thiết kế hệ thống (System Design):

Nhiệm vụ của bước này là xây dựng được sơ đồ triển khai hệ thống (Deployment diagram) nhằm mô tả số lượng, các kiểu và vị trí của các agent trong hệ thống. Dựa vào sơ đồ này, người thiết kế có thể cấu hình hệ thống sao cho tối ưu được sức mạnh tính toán cũng như băng thông của mạng:

- Nếu chú trọng đến chi phí truyền thông, có thể xếp nhiều agent trên một máy tính, nhưng tốc độ xử lý sẽ bị chậm lại và làm mất tính ưu việt của hệ đa agent trong môi trường phân tán.
- Nếu muốn giảm khối lượng tính toán tại mỗi máy, có thể xếp các agent khác nhau trên các máy khác nhau và phải tăng phần chi phí truyền thông.

Mỗi hệ thống con được hiểu là một máy chủ (server) có khả năng hoạt động liên tục trong thời gian chạy ứng dụng và được biểu diễn bằng một hình chữ nhật nét rời. Trong hệ thống con, có thể thiết kế số lượng các agent theo mục đích riêng của người thiết kế. Khi đó, các agent của các lớp agent khác nhau sẽ tương tác với nhau thông qua các phiên hội thoại giữa các lớp agent tương ứng. Các phiên hội thoại này được biểu diễn bằng các mũi tên nét liền theo đúng chiều phiên hội thoại đã thiết kế trước đó.

## 3. AgentTool

# 3.1 Giới thiệu về AgentTool

Hệ thống agentTool là một công cụ để hỗ trợ và thực thi MaSE. Hiện nay AgentTool hỗ trợ tất cả các bước của MaSE cũng như hỗ trợ cho việc chuyển đổi các mô hình phân tích thành các mô hình thiết kế. Các Menu cho phép truy cập tới nhiều chức năng hệ thống, bao gồm một đại diện tri thức dựa trên xác minh cuộc hội thoại và sinh mã. Các nút (Button) cho phép thêm các mục cụ thể tới các biểu đồ và cửa sổ bên dưới hiển thị các thông điệp hệ thống. Các biểu đồ MaSE khác nhau được truy cập thông qua các tab trên cửa sổ chính. Khi một biểu đồ MaSE được chọn, người thiết kế có thể thao tác bằng giao diện đồ họa trong cửa sổ đó. Mỗi bảng có các kiểu khác nhau

của các đối tượng và văn bản mà có thể được đặt trong chúng. Việc lựa chọn một đối tượng trong cửa sổ cho phép các biểu đồ liên quan khác trở thành được liên kết. [1]

Một phần của agentTool mà có lẽ là nổi bật nhất đó là khả năng làm việc trên các bộ phận khác nhau của hệ thống và ở các mức độ trừu tượng thay thế cho nhau. Trong mỗi bước của việc phát triển hệ thống, các biểu đồ thiết kế và phân tích khác nhau là sẵn sàng thông qua các tab trên của sổ chính. Thứ tự của các tab theo các bước của MaSE, vì vậy việc lựa chọn một tab bên trái của bảng hiện hành sẽ di chuyển lùi về một bước của phương pháp luận và việc lựa chọn một tab bên phải sẽ là tiến thêm một bước của phương pháp luận. [1]

## 3.2 Thiết kế các hệ thống sử dụng agentTool

Việc thiết kế một hệ thống multiagent sử dụng agentTool bắt đầu trong một biểu đồ lớp agent. Một cuộc hội thoại chỉ có thể tồn tại giữa các lớp agent, ta phải xác định các lớp agent trước khi xác định các cuộc hội thoại. Ta có thể thêm tất cả các lớp agent vào biểu đồ lớp agent trước khi thêm các cuộc hội thoại, ta cũng có thể thêm các "section" của hệ thống ở một thời điểm, kết nối các lớp agent phù hợp với các cuộc hội thoại và sau đó chuyển sang phần tiếp theo.[1]

Khi đã xác định được các lớp agent và các cuộc hội thoại, ta có thể xác định chi tiết của các cuộc hội thoại bằng cách sử dụng các biểu đồ lớp truyền thông. Nút "Add State" thêm một trạng thái và nút "Add Trans" là thêm một cuộc hội thoại giữa hai trạng thái được lựa chọn. Người thiết kế có thể xác minh một cuộc hội thoại tại bất kỳ điểm nào trong quá trình tạo nó bằng cách sử dụng lệnh Verify Conversations từ menu Verify. Quá trình xác minh đảm bảo rằng chi tiết các cuộc hội thoại là không bế tắc (deadlock free). Nếu bất kỳ một lỗi nào tồn tại, ta sẽ thấy các kết quả xác minh trong một phần được tô đậm của một cuộc hội thoại trên chuyển tiếp "Ack". Mỗi một phần tô đâm chỉ ra một lỗi.[1]

Các lớp agent có các thành phần bên trong mà có thể được thêm vào, được loại bỏ và được thao tác trong một cách tương tự tới các bảng khác. Tuy nhiên, các lớp agent có một lớp phức tạp được thêm vào từ các thành phần có thể có các biểu đồ trạng thái thành phần và có thể thêm các thành phần con. [1]

Người thiết kế có thể thêm thông tin thiết kế chi tiết ở một cấp độ thấp của sự trừu tượng. Các biểu đồ trạng thái thành phần (Component State Diagram) xác định các hành vi động của các thành phần và các biểu đồ kiến trúc con (Sub-Architecture Diagram) chứa các thành phần bổ xung và bộ liên kết mà xác định các thành phần tiếp theo.[1]

# 3.3 AgentTool hỗ trợ việc thiết kế bán tự động

Để thực hiện quá trình này, người thiết kế phải gán các role cho các lớp agent cụ thể, sau đó người thiết kế có thể áp dụng việc bán tự động truyển đổi thành các mô hình phân tích. Có 3 giai đoạn cơ bản cho việc chuyển đổi: giai đoạn 1: các chuyển đổi cố gắng xác định các sự kiện giao thức trong các task đồng thời. Trong hầu hết các trường hợp này có thể được hoàn thành một cách tự động. Tuy nhiên, trong một vài trường hợp, hệ thống không thể xác định chính xác các giao thức thích hợp cho mỗi sự kiện gửi và nhận. Khi điều này xảy ra, hệ thống yêu cầu người thiết kế một sự lựa chọn

trong các giao thức được đưa ra; *giai đoạn 2*: các máy trạng thái hữu hạn của mỗi thành phần được chú thích để chuẩn bị cho việc triết xuất các cuộc hội thoại. Giai đoạn này tìm thấy vị trí bắt đầu và kết thúc của mỗi cuộc hội thoại và đảm bảo rằng các cuộc hội thoại giữa các agent phù hợp; *giai đoạn 3*: các cuộc hội thoại được triết xuất từ các thành phần bên trong và được đặt trong biểu đồ hội thoại riêng biệt. các cuộc hội thoại được thay thế bởi các lời gọi phương thức mà các máy trạng thái thành phần bên trong vẫn duy trì việc xử lý bên trong và cho phép cuộc hội thoại phối hợp.[1]

#### 3.4 Tính Di động

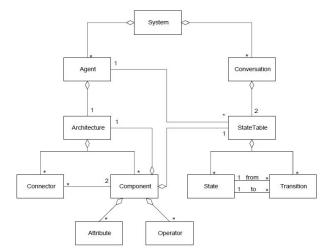
MaSE và agentTool cũng được nghiên cứu để cung cấp các khả năng cho mô hình agent di động. Bước đầu tiên trong việc mô hình hóa tính di động là bao gồm khả năng di chuyển trong một trạng thái task đồng thời. Các hoạt động di chuyển về cơ bản là yêu cầu các agent di chuyển tới một vị trí mới. Việc thực hiện thực tế của các hoạt động di chuyển được giả định là một phần của các môi trường mà trong đó các agent thực thi. Các hoạt động trả về một giá trị Boolean như các di chuyển thực sự được xảy ra. Ngoài ra, pha phân tích cho phép người phân tích xác định khi một di chuyển xảy ra, vị trí được yêu cầu và khả năng để quyết định việc di chuyển có hoặc không thành công. Trong pha phân tích, tính di động phức tạp hơn pha thiết kế. Ở mức độ thiết kế, MaSE đã cung cấp khả năng thông báo cho mỗi thành phần khi một di chuyển được yêu cầu và cung cấp khả năng cho mỗi thành phần để lưu trữ trạng thái hiện thời, tắt và khởi động lại sau mỗi di chuyển. Để giúp cho người thiết kế thực hiện các hoạt động thiết kế phức tạp này, các chuyển đổi bán tự động như được mô tả ở phần trên đã được phát triển và thực hiện trong agentTool.[1]

#### 3.5 Các hình thức agentTool cơ bản

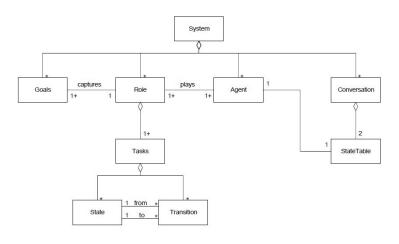
Các ngữ nghĩa chính thức của MaSE được phản ánh trong các chuyển đổi từ một sự trừu tượng kế tiếp. Các ngữ nghĩa này được thành lập và thực thi bởi agentTool. Một phiên bản trong tương lai của agentTool trong đó có sự kết hợp toàn bộ của phương pháp luận MaSE, một role có thể được ánh xạ ngược lại tới tập hợp các Goal mà từ đó nó đã được tạo hoặc chuyển tiếp tới các lớp agent mà nó đảm nhiệm. Hệ thống agentTool được dựa trên một sự phân cấp đối tượng mà trong đó các đối tượng bắt chước các đối tượng trong MaSE. Trong agentTool, mỗi hệ thống được bao gồm một tập hợp của các Agent và các cuộc hội thoại. Mỗi agent có thể có một kiến trúc, trong đó được bao gồm các thành phần và các bộ liên kết. Tương tự như vậy, một cuộc hội thoại được bao gồm hai bảng trạng thái, trong đó bao gồm một tập hợp của các trạng thái (State) và các chuyển tiếp (Transitions). Mô hình đối tượng bên trong của agentTool chỉ cho phép các cấu hình được cho phép bởi MaSE, và nó chính thức thực thi các cấu trúc biểu đồ MaSE và các mối tương quan.

Trong tương lai, mô hình đối tượng agentTool sẽ được mở rộng để kết hợp các Goal, các Role và các Task. Trong mô hình đối tượng mới một agent đóng vai trò ít nhất một Role, trong đó bao gồm một Task hoặc nhiều hơn. Tương tự như vậy, tất cả các Role phải được đóng vai trò ít nhất bởi một agent. Mỗi role cũng xác định được một hoặc nhiều hơn các Goal và mỗi Goal được xác định bởi chính xác một Role. Với mô hình đối tượng dễ ràng nhận thấy rằng nếu người dùng lựa chọn một Agent cụ thể

và không khó khăn để xác định chính xác các Role, các Task, các Goal và các cuộc hội thoại có thể bị ảnh hưởng bởi bất kỳ sự thay đổi nào tới Agent đó. Điều quan trọng hơn là người dùng có thể lựa chọn một Goal và dễ dàng xác định các Role, các Task, các Agent và các cuộc hội thoại có thể bị ảnh hưởng bởi sự thay đổi tới các Goal.[4]



Hình 3: Mô hình đối tượng MaSE hiện tại



Hình 4: Mô hình đối tượng MaSE được mở rộng

# 4. Áp dụng phân tích và thiết kế hệ hỗ trợ dịnh vụ mua và bán điện thoại di động

# 4.1 Giới thiệu hệ hỗ trợ dịch vụ mua và bán máy điện thoại di động

Mô tả bài toán: "Một khách hàng muốn đặt mua một chiếc điện thoại di động. Khách hàng đưa ra các thông tin yêu cầu cụ thể về sản phẩm như: hãng sản xuất, giá, chíp, dung lượng bộ nhớ, thời lượng pin, trọng lượng máy, chức năng, phụ kiện v.v... Khi nhận được yêu cầu từ phía khách hàng, hệ thống sẽ trả lại kết quả phù hợp nhất với thông tin mà khách hàng đã đưa ra. Trong trường hợp kết quả đưa ra chưa thực sự thỏa mãn yêu cầu của khách hàng, có thể tiến hành thương lượng để đạt được kết quả tối ưu nhất."

Xác định yêu cầu: từ mô tả bài toán ở trên, có thể xác định các yêu cầu đối với hệ thống như sau:

- Nhiệm vụ chính của hệ thống là thương lượng để tìm kiếm sản phẩm phù hợp nhất cho khách hàng và đặt mua sản phẩm đó.
- Kết quả trả về phải phù hợp với yêu cầu mà khách hàng đưa ra ban đầu và trong một khoảng thời gian cho phép.
- Trong quá trình hoạt động của hệ thống, khách hàng có thể thay đổi yêu cầu của mình. Do đó, hệ thống cần phải có khả năng đáp ứng đối với những thay đổi này và cập nhật lại yêu cầu của khách hàng.
- Các đại lý bán hàng phải có khả năng quản lý được thông tin về sản phẩm và liên lạc được với khách hàng khi quá trình giao dịch thành công.

# 4.2 Phân tích hệ thống

Theo phương pháp luận MaSE, pha phân tích được thực hiện theo bốn bước như sau:

- Xác định Goal hệ thống
- Xác đinh use case
- Xây dựng Ontology
- Hoàn thiện Role

#### 4.2.1 Xác định Goal

Goal là một khái niệm để chỉ mục đích mà hệ thống cần đạt được. Mục đích của hệ thống ở đây được nhìn từ quan điểm của hệ thống nghĩa là các dịch vụ mà hệ thống có thể cung cấp. Nhiệm vụ của bước này là chuyển toàn bộ yêu cầu của hệ thống thành tập các Goal có cấu trúc của hệ thống. Việc xây dựng goal hệ thống được thực hiện qua hai bước con đó là: xác định Goal và tổ chức Goal

# Xác định goal

Như mô tả yêu cầu ở trên, nhiệm vụ chính của hệ hỗ trợ dịch vụ mua điện thoại di động là tìm kiếm sản phẩm phù hợp nhất với yêu cầu của khách hàng đưa ra và đặt mua sản phẩm đó. Từ hai nhiệm vụ này ta có thể xác định được hai Goal ban đầu của hệ thống là: **Sự thương lượng; Sự đặt mua sản phẩm** 

Sau khi thương lượng, hệ thống phải thông báo kết quả tìm kiếm sản phẩm cho khách hàng. Do đó, ta có thêm Goal: **Sự thông báo kết quả** 

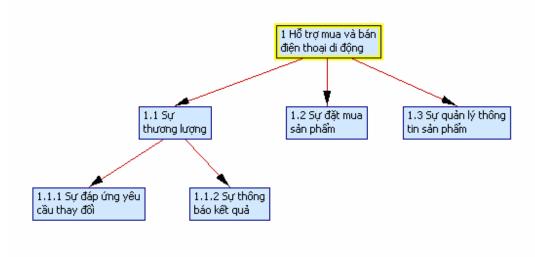
Trong quá trình hoạt động, vì một lí do nào đó khách hàng muốn thay đổi yêu cầu ban đầu của mình, hệ thống phải có khả năng đáp ứng đối với thay đổi đó của khách hàng, do đó ta có Goal sau: **Sự đáp ứng yêu cầu thay đổi** 

Quản lý thông tin về sản phẩm là một yêu cầu cần có của hệ thống thông tin quản lý. Do đó, hệ hỗ trợ dịch vụ mua và bán điện thoại di động cũng cần có khả năng quản lý được thông tin sản phẩm nhằm phục vụ quá trình thương lượng để tìm kiếm sản phẩm phù hợp nhất cho khách hàng, ta có thêm Goal của hệ thống là: **Sự quản lý thông tin sản phẩm** 

Như vậy, các Goal của hệ thống đã xác định được bao gồm: Sự thương lượng; Sự đặt mua sản phẩm; Sự thông báo kết quả; Sự đáp ứng yêu cầu thay đổi; Sự quản lý thông tin sản phẩm

#### ■ Tổ chức cây phân cấp Goal

Trong bước này, các Goal đã được xác định ở bước trước sẽ được tổ chức thành cây phân cấp Goal. Ta có thể nhận thấy các Goal **Sự thương lượng; Sự đặt mua sản phẩm** và **Sự quản lý thông tin sản phẩm** là các mục đích riêng biệt của hệ thống. Khi quá trình thương lượng thành công, hệ thống phải thông báo kết quả này cho người dùng, do vậy có thể đặt Goal **Sự thông báo kết quả** là Goal con của Goal **Sự thương lượng.** Ta cũng có thể thấy, trong quá trình thương lượng, khách hàng có thể thay đổi yêu cầu của mình, vì thế, Goal **Sự đáp ứng yêu cầu thay đổi** có thể là con của Goal **Sự thương lượng.** Như vậy ta có tổ chức các goal như sau:



Hình 5: Biểu đồ phân cấp goal

#### 4.2.2 Xác định Use case

Mục đích của bước này là tạo ra một tập các use case và các sơ đồ tuần tự (Sequence diagram) tương ứng nhằm hỗ trợ cho người phân tích hệ thống phát hiện được tập các role ban đầu và các đường truyền thông có thể có trong hệ thống. Quá trình xây dựng use case gồm hai bước con là: tạo các use case và xây dựng biểu đồ tuần tư.

#### ■ Tao các use case

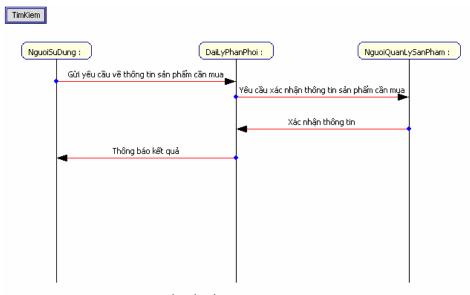
Các use case được xác định dựa trên các yêu cầu đối với hệ thống, là các chức năng mà hệ thống cần phải thực hiện để đáp ứng yêu cầu đối với nó. Việc xác định ra các use case có thể thực sự suy luận ra nhiều thông tin hơn về các Goal của hệ thống. Nói cách khác, để thực hiện được một Goal của mình, hệ thống cần phải thực hiện các chức năng cụ thể để hoàn thành Goal đó. Như vậy, việc xác định các use case không chỉ dựa trên các yêu cầu chức năng của hệ thống mà còn có thể xác định bằng cách trả lời các câu hỏi là làm thế nào để hệ thống thực hiện được một Goal cụ thể trong biểu đồ phân cấp Goal.

Trong hệ thống hỗ trợ dịch vụ mua bán mua bán điện thoại di động, có thể xác định được các use case như sau: CapNhatYeuCau: được trích xuất ra từ yêu cầu đáp ứng yêu cầu thay đổi của khách hàng. Trong quá trình hoạt đông, khách hàng muốn thay đổi yêu cầu của mình, hệ thống cần phải thực hiện cập nhật lại yêu cầu của khách hàng và tiến hành lại cuộc thương lượng; **ThuongLuong**: khi hệ thống tìm kiếm sản phẩm không có kết quả chính xác theo mong muốn của khách hàng, các bên có thể thương lượng với nhau để đi đến một kết quả tối ưu nhất; **TimKiem:** trong quá trình thương thương, hệ thống cần phải tìm các sản phẩm phù hợp nhất với yêu cầu của khách hàng; HienThiKetQua: Quá trình thương lượng thành công hay không thì hệ thống vẫn phải thực hiện thông báo kết quả cho khách hàng; **DatHang:** Khi đã tìm được sản phẩm phù khách hàng hiên chức hop. thuc năng đăt **CapNhatThongTinSanPham:** Những thông tin về sản phẩm luôn luôn được cập nhật: thêm sản phẩm, sửa một số thông tin về sản phẩm và có thể thực hiện xóa sản phẩm khi sản phẩm đó đã được bán. Như vậy, các use case của hệ thống là: CapNhatYeuCau; TimKiem; HienThiKetQua; ThuongLuong; DatHang; CapNhatThongTinSanPham

# Xây dựng biểu đồ tuần tự

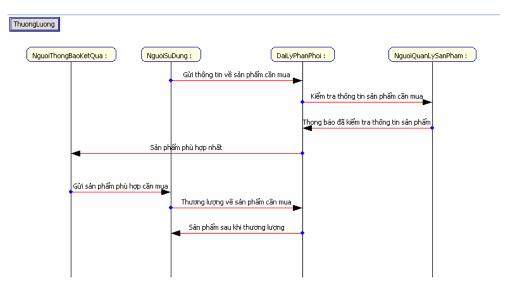
Xây dựng các biểu đồ tuần tự là xây dựng các giao tiếp giữa các role trong hệ thống. Để xác định các role, ta có thể sử dụng kỹ thuật trích danh từ. Từ các use case đã có trong bước trước, người phát triển sẽ tiến hành duyệt và tìm ra các danh từ chỉ các đối tượng có chức năng cụ thể nào đó trong use case trước đó. Tiếp theo, ta sẽ xem xét lại các danh mục các danh từ này và loại đi các danh từ chỉ cùng một đối tượng. Các danh từ còn lại sẽ là các role sẽ có mặt trong sơ đồ tuần tự tương ứng với use case đó. Sau khi đã có các role thì trong bước tiếp theo ta sẽ tiến hành biểu diễn tuần tự các sự kiện của sơ đồ tuần tự trong AgentTool.

Ta có các role của hệ thống là: **DaiLyPhanPhoi;NguoiDapUngThayDoi**; **NguoiThongBaoKetQua;NguoiDatHang;NguoiQuanLySanPham;NguoiSuDung**Từ các role ở trên ta có các biểu đồ tuần tư cho các use case như sau:

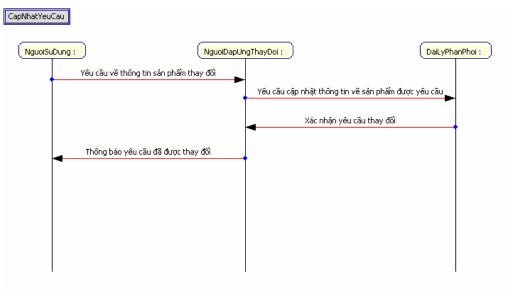


Hình 6: Biều đồ tuần tự của use case TimKiem

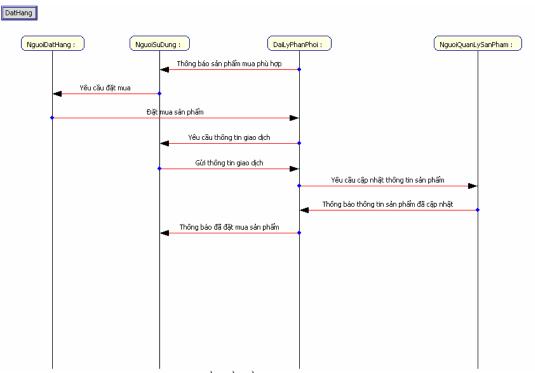
# Công nghệ phần mềm hướng Agent



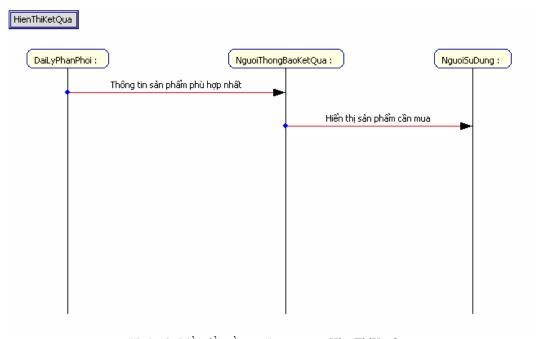
Hình 7: Biểu đồ tuần tự của use case ThuongLuong



Hình 8: Biều đồ tuần tự của use case CapNhatThayDoi



Hình 9: Biều đồ tuần tự của use case DatHang



Hình 10: Biều đồ tuần tự của use case HienThiKetQua

#### 4.2.3 Xây dựng Ontology

Theo phương pháp luận MaSE có bốn bước chính để xây dựng nên ontology cho một ứng dụng đó là: *xác định mục đích và phạm vi của ontology, thu thập dữ liệu, xây dựng ontology ban đầu* và bước cuối cùng là *hoàn thiện và kiểm định ontology*.

Xác định mục đích và phạm vi của Ontology

Việc xác định mục đích và phạm vi của ontology dựa vào các yêu cầu chức năng và cấu trúc goal phân cấp đã được xây dựng ở trong bước đầu tiên của phương pháp luận MaSE. Xác định được mục đích và phạm vi của ontology cho phép ta nhìn nhận một cách nhanh chóng về những thông tin được chứa trong ontology.

Trong hệ thống hỗ trợ dịch vụ mua và bán điện thoại di động, mục đích chính là để tìm kiếm sản phẩm đúng theo yêu cầu khách hàng. Trong trường hợp không tìm thấy chính xác sản phẩm được yêu cầu, khách hàng và đại lý phân phối có thể tiến hành thương lượng để đưa ra một kết quả tối ưu nhất cho khách hàng. Như vậy, cần phải có thông tin về sản phẩm, thông tin về khách hàng, thông tin về đại lý phân phối.

# Thu thập dữ liệu

Sau khi xác định được các loại dữ liệu và mức độ chi tiết của chúng, ta phải lập ra một danh sách liệt kê tất cả các danh từ có mặt trong cây phân cấp Goal (đã có ở bước xác định Goal) và các use case. Danh sách này được gọi là "danh sách thuật ngữ" của hệ thống. Từ tập thuật ngữ này, ta phải chọn ra các danh từ có thể biểu diễn một kiểu dữ liệu cần thiết cho hệ thống bằng cách xem xét tất cả các kiểu dữ liệu cần truyền đi trong các use case. Bước này thực chất là tìm ra các khái niệm có trong các message và các khái niệm ở mức nhỏ hơn để các agent có thể hiểu được các message mà nó nhận được. Sau bước này ta nhận được các khái niệm cần có trong ontology.

Theo cách này, có thể xây dựng được các thuật ngữ và các khái niệm ứng cử của ontology dựa trên việc xây dựng các thuật ngữ, các khái niệm trong miền tri thức về khách hàng, đại lý phân phối và sản phẩm bán.

Các thuật ngữ có liên quan đến thông tin về khách hàng bao gồm: Địa chỉ của khách hàng; Tên khách hàng; Số điện thoại của khách hàng; Địa chỉ mail của khách hàng; Số tài khoản ngân hàng của khách hàng

Các thuật ngữ có liên quan đến thông tin về sản phẩm bao gồm: Hãng sản xuất; Giá của sản phẩm; Chíp: Bộ xử lý của điện thoại di động; Tốc độ xử lý; Dung lượng bộ nhớ trong; Dung lượng bộ nhớ ngoài; Màu sắc; Thời lượng pin; Trọng lượng máy; 3G; GPS; Mp3; MMS; Wi-Fi; 3 Sim; GPRS; Camera; FM radio; Bluetooth; Loa ngoài; Java Game; 2 Sim online; Ứng dụng VP; Âm thanh Hifi; Thẻ nhớ ngoài; Màn hình màu; Màn hình cảm ứng; Bàn phím QWERTY; Thân gập; Thân xoay; Thân trượt; Thân thẳng; Màn hình HD

Thông tin liên quan đến thông tin về đại lý phân phối: Tài khoản

# Xây dựng Ontology ban đầu

Để xây dựng được Ontology khởi đầu, người thiết kế phải chuyển toàn bộ các khái niệm đã thu được trong bước 2 vào tập các lớp và các thuộc tính của chúng. Ta có thể lựa chọn một trong hai cách: *Kế thừa từ một Ontology của một hệ thống khác*; *Xây dựng Ontology ngay từ*. Trong hệ thống này, ontology được xây dựng mới ngay từ đầu. Có thể tổ chức các khái niệm liên quan đến việc hoàn thành các Goal của hệ thống vào các lớp như dưới đây:

# Công nghệ phần mềm hướng Agent

# ChucNang 3G gps mp3 mms wi-fi 3Sim gprs

camera fmradio bluetooth loaNgoai

javaGame 2simOnline

ungDungVP amThanhHifi

theNhoNgoai manHinhMau

manHinhCamUng banPhimQWERTY

thanGap thanXoay thanTruot

thanThang manHinhHD

#### KhachHang

tenKhachHang diaChi email soDienThoai soTaiKHoan

DaiLyPhanPhoi

taiKhoan

#### HangSanXuat

Nokia
Samsung
Q-Mobile
Philips
Mobistar
Avio
P-Phone
LG
Motorola
Sony Ericsson
F-Mobile (FPT)
Gigabyte

Sharp

#### PhuKien

theNho
thietBiKetNoi
pin
sac
nguon
taiNgheBluetooth
taiNgheCoDay
giaDeDTtrenXeHoi
linhKienKhac

#### Chip

hangSanXuat
tocDo
cache
soNhan
congNghe

#### DungLuongBoNho

dungLuongBoNhoTrong dungLuongBoNhoNgoai hangSanXuat gia model

#### SanPham

hangSanXuat
gia
chip
dungLuongBoNho
thoiLuong
trongLuong
chucNang
phuKien

#### Gia

<2trieu
2trieu\_3trieu
3trieu\_4trieu
4trieu\_5trieu
5trieu\_6trieu
>6trieu

#### CauHinh

hangSanXuat chip dungLuongBoNho gia

#### Hoàn thiện và kiểm định Ontology

Trong bước này, ta phải chứng thực được rằng Ontology vừa xây dựng đã thỏa mãn yêu cầu của hệ thống bằng cách xem xét lại toàn bộ các tình huống đã được mô tả trong các use case và sơ đổ tuần tự: Nếu phát hiện thấy một thiếu sót nào đó thì khái niệm tương ứng phải được bổ xung ngay vào Ontology; Nếu phát hiện thấy có khái niệm nào không cần thiết thì phải loại bỏ ngay ra khỏi Ontology

#### 4.2.4 Hoàn thiện Role

Mục tiêu của bước cuối cùng trong pha phân tích là hoàn thiện các role, có nghĩa là chuyển các goal đã được cấu trúc và các biểu đồ tuần tự thành các role thực sự của hệ thống và các nhiệm vụ phối hợp của chúng, đây là dạng phù hợp hơn cho việc thiết kế các MAS. Role là dạng cơ bản cho việc định nghĩa các lớp agent và biểu diễn các goal hệ thống trong suốt pha thiết kế. Các goal hệ thống sẽ thõa mãn nếu mọi goal kết hợp được với role và mỗi role được thực hiện bởi một lớp agent.

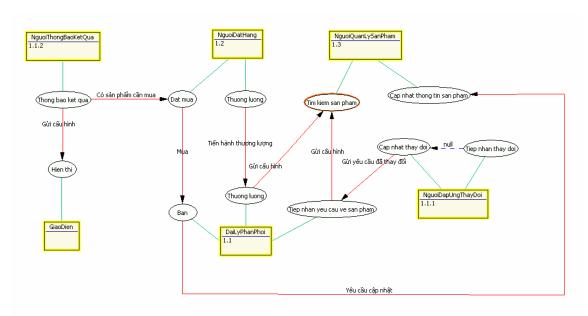
Các trường hợp chuyển đổi thông thường các goal tới các role là kiểu 1-1, với mỗi goal ánh xạ thành một role. Tuy nhiên, có nhiều tình huống mà một role đơn chịu trách nhiệm nhiều goal. Có nhiều sự xem xét trong quá trình hoàn thiện các role. Các goal tương tự hoặc liên quan đến nhau có thể kết hợp thành một role đơn.

Các goal của hệ thống: Sự thương lượng; Sự đặt mua sản phẩm; Sự thông báo kết quả; Sự đáp ứng yêu cầu thay đổi; Sự quản lý thông tin sản phẩm

Ánh xạ thành các role theo hình thức ánh xạ 1 – 1 ta có các role tương ứng trong hệ thống là: **DaiLyPhanPhoi** (1.1):Thực hiện goal **Sự thương lượng**; **NguoiDatHang** (1.2): Thực hiện goal **Sự đặt mua sản phẩm**; **NguoiThongBaoKetQua** (1.1.2):Thực hiện goal **Sự thông báo kết quả**; **NguoiDapUngThayDoi** (1.1.1):Thực hiện goal **Sự đáp ứng yêu cầu thay đổi**; **NguoiQuanLySanPham** (1.3):Thực hiện goal **Sự quản lý thông tin sản phẩm**; **NguoiSuDung** 

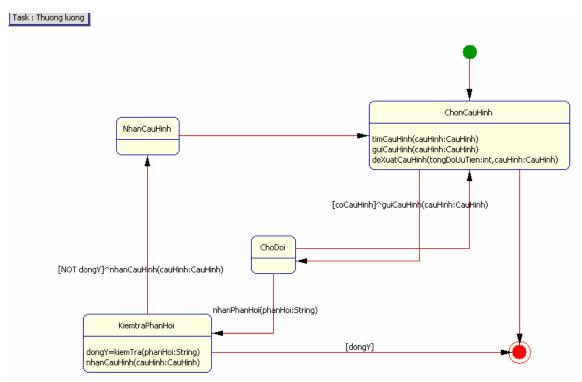
Tiếp theo, ta cần phải xác định các task của các role và các giao tiếp của các role đã được xác định ở trước. Kết quả của bước này là gán được các task cho các role tương ứng của nó. Chẳng hạn, đối với role DaiLyPhanPhoi thực hiện các task là tiếp nhận yêu cầu từ phía NguoiDapUngThayDoi và thương lượng với NguoiDatHang nên có hai task là: Thuong lưong và Tiep nhan yeu cau ve san pham. Tương tự, đối với role NguoiDapUngThayDoi cũng sẽ có hai task là Tiep nhan thay doi và Cap nhat thay doi. Các task của các role được mô tả trong biểu đồ mô hình role như dưới đây:

# Công nghệ phần mềm hướng Agent



Hình 11: Mô hình role

Sau khi xây dựng xong mô hình role, việc tiếp theo là cần phải mô tả các task đồng thời của các role đó. Dưới đây là biểu đồ task đồng thời của các role.



Hình 12: Biểu đồ task Thuong luong của role DaiLyPhanPhoi

# 4.3 Thiết kế hệ thống

## 4.3.1 Tạo lớp agent

Các lớp agent được tạo ra từ các role. Sản phẩm của pha này là biểu đồ lớp agent trong đó mô tả tổ chức tổng thể của hệ thống bao gồm các lớp agent và các cuộc hội thoại giữa chúng. Một lớp agent là một kiểu mẫu của agent trong hệ thống và tương tự đối với một lớp đối tượng trong lập trình hướng đối tượng. Một agent là một thể hiện của lớp agent. Trong suốt pha này, các lớp agent được xác định theo các role mà chúng sẽ đảm nhiệm và các cuộc hội thoại trong đó chúng phải tham dự.

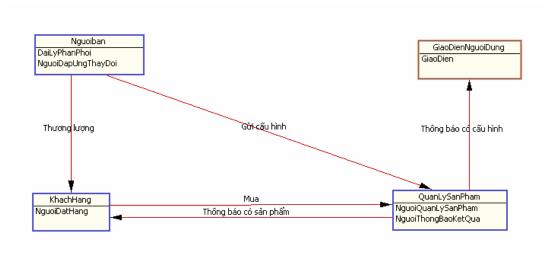
Role chính là cơ sở để xây dựng nên các lớp agent. Do đó, để đảm bảo tất cả các goal của hệ thống được thực hiện trong thiết kế, phải có ít nhất một lớp agent được gán cho mỗi role mà đã được xác định trong pha phân tích. Trong thực tế, các lớp agent có thể thực hiện nhiều role, với các role thay đổi một cách tự động trong quá trình thực hiện. Hơn nữa, các agent của cùng một lớp agent có thể thực hiện các role khác nhau tại cùng một thời điểm.

Cũng trong pha này, cần phải cũng xác định các cuộc hội thoại giữa các agnet tham gia. Tập các hội thoại của một lớp agent tham gia vào đó được dẫn xuất từ sự giao tiếp bên ngoài của các role mà agent đảm nhiệm.

Các lớp agent và các cuộc hội thoại được tài liệu hóa thông qua biểu đồ lớp agent. Biểu đồ này tương tự với biểu đồ lớp trong hướng đối tượng, nhưng có hai điểm khác nhau cơ bản:

- Các lớp agent không được định nghĩa bởi các thuộc tính và các phương thức, mà chúng được định nghĩa bởi các role mà chúng thực hiện
- Ngữ nghĩa quan hệ giữa các lớp agent. Tất cả các quan hệ giữa các lớp là các cuộc hội thoại có thể đặt giữa 2 agent.

Theo đó, để thiết kế hệ hỗ trợ dịch vụ mua bán điện thoại di động cần thực hiện xác định ra các agent thực hiện các role trong hệ thống và mô tả các cuộc hội thoại giữa những agent đó. Các agent trong hệ thống này là: NguoiBan: Thực hiện các role DaiLyPhanPhoi và NguoiDapUngThayDoi; KhachHang: Thực hiện role NguoiDatMua; QuanLySanPham: Thực hiện role NguoiQuanLySanPham và ThongBaoKetQua; GiaoDienNguoiDung: Thực hiện role GiaoDien



Hình 13: Biểu đồ lớp agent

Tất cả các giao thức giữa các task trong mô hình role đều trở thành các conversation. Các tương tác bên trong, chẳng hạn như hai task Tiep nhan thay doi và Cap nhat yêu cau không trở thành conversation vì nó được thực hiện ở ngay chính bản thân role NguoiDapUngThayDoi. Như vậy, ta có các conversation như sau: Gui Cau Hinh; Thuong Luong; Mua; Thong bao co san pham; Thong bao co cau hình

#### 4.3.2 Xây dựng các conversation

Nhiệm vụ của bước này là thiết kế chi tiết kiến trúc bên trong của các phiên hội thoại đã được xác định trong bước xác định các lớp agent. Đối với mỗi phiên hội thoại phải làm sáng tỏ được cách thức và cơ chế hoạt động của mỗi bên agent tham gia vào phiên hội thoại đó.

Trong bước xác định các lớp agent, các phiên hội thoại đã được xác định từ các tương tác giữa các role của các lớp agent khác nhau. Do vậy các sơ đồ lớp truyền thông cũng được xác định tương ứng với các task mà các role đó thực hiện. Thông thường, mỗi sơ đồ task sẽ tương ứng với một sơ đồ phiên hội thoại cho bên tham gia tương ứng.

Tại các trạng thái của phiên hội thoại, ta phải xác định chính xác tên các hàm, các biến và các tham số cho các hàm.

Tại các chuyển tiếp của sơ đồ phiên hội thoại, ta phải chi tiết hóa dựa trên các chuyển tiếp trong sơ đồ task tương ứng. Cùng dựa trên cú pháp của chuyển tiếp <receive (message)><[guard]><send(message)> ,nhưng được cụ thể hóa nội dung các thông điệp (message) dựa vào việc sử dụng Ontology của hệ thống đã được thiết kế trong bước xây dựng Ontology.

Với hệ thống này ta có các conversation như sau: Conversation Initiator: Thương lượng; Conversation Responder: Thương lượng; Conversation Initiator: Gửi cấu hình; Conversation Responder: Gửi cấu hình; Conversation Initiator: Thông báo có cấu hình; Conversation Responder: Thông báo có cấu hình; Conversation Initiator: Mua; Conversation Responder: Mua; Conversation Initiator: Thông báo có sản phẩm; Conversation Responder: Thông báo có sản phẩm;

#### 4.3.3 Hoàn thiện các agent

Bao gồm hai bước con: thiết kế kiến trúc bên trong agent và thiết kế các thành phần trong kiến trúc đó.

#### Thiết kế kiến trúc

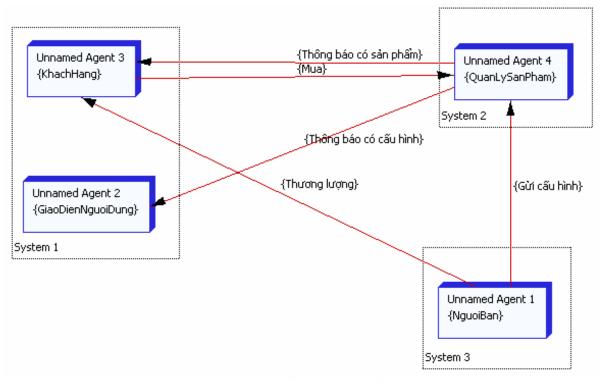
Được xác định từ các Task của các Role mà lớp Agent tương ứng đảm nhiệm. Như vậy, với hệ thống này ta có thể xây dựng các thành phần sau: Agent **NguoiBan** có các role là: **DaiLyPhanPhoi** và **NguoiDapUngThayDoi** nên ta có các thành phần chính là các Task của các role này như sau: **Ban**, **Thuong luong**, **Tiep nhan yeu cau ve san pham**, **Tiep nhan thay doi**, **Cap nhat thay doi**. Tương tự, Agent **KhachHang** có các thành phần: **Dat mua**, **Thuong luong**; Agent **GiaoDienNguoiDung** có các thành phần: **Hien thi**; Agent **QuanLySanPham** có các thành phần: **Tim kiem san pham**, **Cap nhat thong tin san pham**, **Thong bao ket qua** 

# ■ Thiết kế các thành phần

Phải phù hợp với sơ đồ role của hệ thống đã được xác định trước đó tức là quan hệ giữa các thành phần phải thống nhất với quan hệ giữa các Task trong sơ đồ Role. Như vậy với Agent **NguoiBan** ta có thành phần **Cap nhat thay doi** tương tác với **tiep nhan yeu cau ve san pham** vì đó cũng chính là 2 Task tương tác với nhau trong sơ đồ Role

# 4.3.4 Thiết kế hệ thống

Nhiệm vụ của bước này là xây dựng được sơ đồ triển khai hệ thống nhằm mô tả số lượng, các kiểu và vị trí của các agent trong hệ thống. Với hệ thống này ta có sơ đồ triển khai hệ thống như sau (Hình 14): Agent **KhachHang** và **GiaoDienNguoiDung** sẽ được xếp trên hệ thống System1; Agent **QuanLySanPham** được xếp trên hệ thống System2; Agent **NguoiBan** được xếp trên hệ thống System3. Và các Agent tương tác với nhau thông qua các phiên hội thoại đã được xác định trước đó



Hình 14: Biểu đồ triển khai hệ thống

#### 5. Đánh giá phương pháp luận MaSE

Để đánh giá phương pháp luận MaSE ta dựa theo các phạm vi mà MaSE đề cập [5]

#### 5.1 Các khái niệm và các thuộc tính

#### ■ Tính tư chủ

Trong Mase tính tự chủ được thể hiện bởi các role đóng gói các chức năng của nó. Chức năng này (ví dụ như các task của nó) là bên trong và nó không bị ảnh hưởng bởi môi trường, do đó đại diện cho tính tự chủ của role.

#### Tính phản ứng

Trong mase tính phản ứng không được thể hiện một cách rõ ràng. Đó là, không có mỗi liên hệ rõ ràng giữa các sự kiện và các hành động được thực hiện. tuy nhiên, tính phản ứng có thể được thể hiện bằng cách sử dụng các máy trạng thái cuộc hội thoại.

#### Tính chủ đông

Trong mase tính chủ động được thể hiện bởi các task của các role. Những nhiệm vụ này (task) được mô phỏng bằng cách sử dụng trạng thái hữu hạn automata.

#### Tính xã hôi

Trong mase các khía cạnh xã hội của một hệ thống (ngoại trừ cho việc truyền thông) không được đề cập. MaSE không cung cấp phương tiện để thu thập các agent hoặc xác định các tổ chức hoặc các xã hội. Tuy nhiên một số khía cạnh xã hội có thể được xem xét bằng cách sử dụng các quy tắc tổ chức.

#### 5.2 Các ký hiệu và các kỹ thuật mô hình hóa

#### Khả năng truy cập

MaSE cung cấp một tập hợp rất đơn giản của các mô hình, mà trong đó tăng cường khả năng truy cập, tuy nhieenm nhu cầu đồng bộ hóa giữa các mô hình và sự cần thiết để thực hiện một mô hình chuyển đổi trong suốt giai đoạn phát triển làm giảm khả năng truy cập của nó.

# Khả năng phân tích

MaSE hỗ trợ kiểm tra tính thống nhất và xác mình nội bộ của các mô hình. Tuy nhiên, vẫn còn một số trường hợp không nhất quán có thể xảy ra. Ví dụ: giữa một biểu đồ tuần tự và các chuyển tiếp giữa các trạng thái trong một cuộc hội thoại.

# Quản lý độ phức tạp

Có một vài lớp của sự trừu tượng trong MaSE: các agent, role và các task. Tuy nhiên, không có hỗ trợ của việc quản lý độ phức tạp của các role và các task phức tạp, ví dụ: không có các phương tiện để mô tả hợp phần của các task. Ngoài ra, MaSE không cho phép xác định một hệ thống phân cấp role.

#### • Khả năng thực thi

MaSE hỗ trợ một phần việc sinh mã bằng cách sử dụng agentTool. Việc sinh mã bao gồm việc hoàn thành các cuộc hội thoại cho nhiều khuôn khổ giao tiếp. MaSE dựa trên trạng thái hữu hạn automaton nó có khả năng để đạt được một việc sinh mã chất lượng cao.

#### Tính biểu cảm

- Cấu trúc của một hệ thống được mô tả rõ ràng bằng cách sử dụng các kiến trúc agent và các biểu đồ thiết kế hệ thống
- Các tri thức được đóng gói trong hệ thống không được trình bày một cách rõ ràng
- Các ontology của hệ thống được hỗ trợ đầy đủ
- Các luồng dữ liệu trong hệ thống không thể được xác định rõ ràng
- Các luồng điều khiển trong hệ thống không được trình bày rõ ràng. Tuy nhiên nó có thể được hiểu từ tập hợp các biểu đồ task đồng thời.
- Kiến trúc vật lý của hệ thống được quy định cụ thể bằng cách sử dụng các biểu đồ triển khai
- Tính di động của các agent được đề cập trong MaSE thông qua một phương pháp đặc biệt (ví dụ như: các phương thức di chuyển)
- Các tương tác của hệ thống với các hệ thống bên ngoài được quy định cụ thể bằng cách sử dụng khái niệm cuộc hội thoại
- Các đặc điểm giao diện người dùng không được đề cập, tuy nhiên MaSE khuyến cáo rằng giao diện người dùng sẽ được xử lý như là một role riêng biệt

#### Tính Modun

Trong MaSE được hỗ trợ trong các biểu đồ mẫu agent, tuy nhiên, việc tái sử dụng các yếu tố trong MaSE chẳng hạn như các task, các role, các giao thức, các cuộc hội thoại là được hỗ trợ.

#### Tính chính xác

Các ngữ nghĩa trong MaSE rõ ràng và do đó ngăn chặn sự hiểu sai bởi các người sử dụng nó. Tuy nhiên, MaSE không cung cấp các định nghĩa chính thức của các khái niêm và các mô hình của nó

# 5.3 Quá trình phát triển

# Bối cảnh phát triển

MaSE là phù hợp cho các bối cảnh phát triển sau: nó có thể được sử dụng trong việc tạo ra phần mềm mới, tái cấu trúc, và việc thiết kế các hệ thống với các thành phần tái sử dụng và tạo mẫu. Tuy nhiên, MaSE không hỗ trợ các kỹ thuật ngược lại vì nó có thể khó chuyển đổi các mô hình về sau

# Phạm vi vòng đời

Là toàn diện trong MaSE. Các bước con của việc xác đinh Goal và xác định use case có thể được xem xét như là một giai đoạn yêu cầu, giai đoạn phân tích bao gồm các bước con của việc xác định các role và các task và các bước con của việc xây dựng hệ thống Ontology, giai đoạn thiết kế bao gồm các bước con của việc xây dựng và hoàn thiện các agent và giai đoạn thực hiện bao gồm các bước con của việc thiết kế hệ thống và sinh mã. Bước thử nghiệm không được xác định trong MaSE

#### Các hoạt động của các giai đoạn trong phương pháp luận

MaSE cung cấp các hướng dẫn cho việc thực hiện các hoạt động của các giai đoạn phát triển.

#### Xác minh và xác nhận

MaSE thực hiện kiểm tra trên các mô hình của nó để kiểm tra tính nhất quán, xác định các bế tắc và các yếu tố không được sử dụng. Ngoài ra, nó cung cấp các hướng dẫn mà hỗ trợ phạm vi kiểm tra giữa các giai đoạn. tuy nhiên, MaSE không cung cấp các hướng dẫn hoặc các phương tiện cho việc kiểm tra các yêu cầu đối với các kết quả của mỗi một trong các giai đoạn đó

#### Đảm bảo chất lượng

Vấn đề này không được đề cập trong MaSE

# Các hướng dẫn quản lý dự án

Vấn đề này không được đề cập trong MaSE

#### 5.4 Thực tế

#### Tài nguyên

MaSE đã có nhiều xuất bản. nó cũng có một website và một công cụ CASE (Agenttool), tuy nhiên không có các nhóm người dùng, cũng không đào tạo hoặc các dịch vụ được cung cấp.

#### Yêu cầu chuyên môn

MaSE đòi hỏi một nền tảng vững chắc và kiến thức trong logic và thời gian logic cho việc sử dụng các quy tắc tổ chức. các mô hình khác không đòi hỏi kiến thức cụ thể ngoại trừ cho trạng thai hữu hạn automata.

#### Khả năng phù hợp ngôn ngữ

MaSE không được nhằm mục tiêu vào một ngôn ngữ lập trình cụ thể, một kiến trúc cụ thể, hoặc một framework cụ thể.

#### Khả năng ứng dụng miền

MaSE là một phương pháp có mục đính chung cho việc thiết kế MAS. Các nhà thiết kế của MaSE báo cáo về việc sử dụng MaSE cho nhiều kiểu của các Agent và các domain.

#### Khả năng mở rộng

MaSE không cung cấp các chi tiết liên quan đến việc sử dụng các tập con hoặc một tập cha cho việc phát triển hệ thống. Nó giống như khi sử dụng MaSE đề xác định một hệ thống của kích thước bất kỳ, người sử dụng nó phải tuân theo con đường phát triển thiết lập bởi MaSE.

## 6. Kết luận

MaSE là một quá trình gồm 8 bước mà biến đổi một tập hợp các mô hình trừu tượng vào một loạt các đại diện cụ thể hơn. MaSE bắt đầu trong pha phân tích bằng cách xác định bản chất của một bối cảnh hệ thống ban đầu trong một tập hợp cấu trúc của các Goal và các use case. Tiếp theo, các use case được truyển đổi thành các biểu đồ tuần tự, bởi vậy các chuỗi sự kiện được mong muốn sẽ được thiết kế vào hệ thống. Cuối cùng, các role được xác định từ các Goal và các use case và bao gồm các task, trong đó mô tả liên kết các Goal được thỏa mãn. Mục đích của pha thiết kế là xác định tổ chức của toàn hệ thống bằng cách chuyển các role và các task được xác định trong pha phân tích thành các kiểu agent và các cuộc hội thoại. Khi một cấu trúc tổ chức hệ thống được xác định, cấu trúc bên trong của mỗi lớp Agent được xác định. Cấu hình hệ thống cuối cùng được được xác định trong bước thiết kế hệ thống. [6]

MaSE và agentTool là các công trình tiến bộ. MaSE cùng với phiên bản hiện tại của agentTool đã được sử dụng để phát triển hơn một chục hệ thống Multiagent khác nhau với hơn một trăm Agent. Người sử dụng cho rằng MaSE sử dụng tương đối đơn giản, nhưng là đủ linh hoạt để cho phép một loạt các giải pháp. Ngoài ra MaSE và agentTool còn đang được sử dụng để phát triển các hệ thống multiagent với quy mô lớn hơn và cũng đang được mở rộng để để xử lý các hệ thống di động và các hệ thống năng động (trong điều kiện các agent có thể nhập và rời khỏi hệ thống trong quá trình thực thi). Ngoài ra MaSE cũng đang được tìm hiểu kỹ hơn về mối quan hệ giữa các task, các cuộc hội thoại và các thiết kế bên trong của các agent. Thêm hai vấn đề đòi hỏi phải nghiên cứu đó là sử dụng những trở ngại tới các Goal như là một cách để tăng khả năng xử lý ngoại lệ và mô hình hóa của các miền thông tin. AgentTool cũng đang được mở rộng để xử lý tất cả các pha và các bước của MaSE bao gồm cả việc sinh mã, và đang được tìm kiếm các kỹ thuật trực quan để thực hiện các biểu đồ MaSE hiện tại hoặc được sửa đổi các phiên bản của chúng để dễ dàng hơn trong việc xem và sử dụng AgentTool. [6]

#### Tài liệu tham khảo

- [1] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.21.2907&rep=rep1&type=pdf
- [2] http://www.pa.icar.cnr.it/cossentino/al3tf1/docs/mase4agentlink.pdf
- [3] http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA448191

# Công nghệ phần mềm hướng Agent

- $[4] \ http://www.springerlink.com/index/0pc24utnubkm51u1.pdf$
- [5]http://ebookee.org/Methodologies-and-Software-Engineering-for-Agent-Systems-The Agent-Oriented-Software-Engineering-Handbook\_123127.html
- [6] http://people.cis.ksu.edu/~sdeloach/publications/Journal/MaSE%20-%20IJSEKE.pdf