

TƯƠNG TÁC DỰA TRÊN ROLE TRONG HỆ ĐA AGENT

Trần Đình Quế

Khoa Công nghệ Thông tin
Học viện Công nghệ Bưu chính Viễn thông
Email: tdque@ptithcm.edu.vn

Nguyễn Thị Thanh Huyền

Trường Đại học Công nghệ
Đại học Quốc gia Hà Nội
Email: huyenntt_dhcn@vnu.edu.vn

Tóm tắt: Các môi trường tính toán số hiện thời thường là không thuần nhất, động và phân tán. Việc phát triển các hệ phần mềm phức tạp trên môi trường như vậy đòi hỏi phải có các phương pháp luận và công cụ phù hợp. Mặc dù phương pháp hướng đối tượng đã được sử dụng rộng rãi hiện nay nhưng nó tỏ ra không thích ứng với yêu cầu thích nghi, phân tán và tương tác linh hoạt. Trong những năm gần đây, cách tiếp cận hướng agent đã thu hút nhiều quan tâm nghiên cứu và được cho là thích hợp cho phát triển các hệ phần mềm phức tạp. Sự thích ứng của hệ đa agent với môi trường tính toán ngày nay là do khả năng tương tác linh hoạt giữa các agent tự chủ khi thực hiện một công việc vượt quá khả năng của mình. Trong các mô hình tương tác cho các hệ đa agent, mô hình tương tác dựa theo cấu trúc xã hội với role đã thu hút nhiều quan tâm nghiên cứu. Mục đích của bài báo này là nhằm trình bày mô hình tương tác dựa trên khái niệm role và xem xét áp dụng mô hình này cho thương lượng song phương trong thương mại điện tử.

Từ khoá: Hệ đa agent, công nghệ phần mềm hướng agent, tương tác, role, thương lượng.

1. GIỚI THIỆU

Các môi trường tính toán hiện thời như mạng truyền thông, mạng Internet, web ngữ nghĩa (semantic web), tính toán hầu khắp (pervasive/ubiquitous computing)...thường không đồng nhất, phân tán, động, mở mà chúng ta không thể đoán trước các hành vi của hệ thống trong thời gian thiết kế ([1], [2]). Việc phát triển các hệ phần mềm phức tạp trên môi trường như vậy đòi hỏi phải có phương pháp luận và công cụ phù hợp. Mặc dù phương pháp hướng đối tượng được sử dụng rộng rãi hiện nay là một tiến hoá đáng kể trong lĩnh vực kỹ nghệ phần mềm của những năm 80-90 nhưng nó tỏ ra không thích ứng với yêu cầu thích nghi, phân tán và tương tác linh hoạt ([1]). Trong những năm gần đây, cách tiếp cận dựa trên agent cho phát triển các hệ phần mềm phức tạp đã thu hút nhiều quan tâm nghiên cứu ([1-5]). Ngoài khả năng hướng đích, phân xạ một cách tự chủ với sự thay đổi của môi trường, các thành phần gọi là agent còn có khả năng tương tác với nhau hoặc dưới dạng cộng tác, hợp tác, cạnh tranh, thương lượng [5].

Tương tác trong hệ đa agent càng ngày càng được xem là một vấn đề quan trọng cần phải đối mặt khi phát triển các ứng dụng phân tán.

Cho đến nay có nhiều mô hình tương tác đã được đề nghị như dựa trên đích hay dựa trên role và mô hình dựa trên role¹ được xem là có nhiều ưu điểm trong phát triển phần mềm ([3], [4]). Mục đích của bài báo này là nhằm trình bày mô hình tương tác dựa trên role và những vấn đề quan tâm nghiên cứu. Bài báo được cấu trúc như sau: Phần 2 trình bày một số đặc trưng tương tác trong hệ hướng đối tượng truyền thống; Phần 3 trình bày mô hình tương tác trong hệ đa agent; Phần 4 là một ví dụ minh họa cho dạng tương tác song phương dựa trên role và Phần 5 là kết luận.

2. TƯƠNG TÁC TRONG HỆ HƯỚNG ĐỐI TƯỢNG

Hiện nay, các phương pháp kỹ nghệ phần mềm như hướng đối tượng, hướng thành phần hay hướng agent đều có xu hướng phân rã hệ thống muốn phát triển thành các thành phần nhỏ hơn. Mỗi thành phần này sẽ thực hiện một nhiệm vụ nào đó và tổng hợp công việc của tất cả các thành phần này sẽ cho kết quả tổng thể của hệ thống. Như vậy, hai điều quan trọng mà chúng ta cần phải quan tâm khi xây dựng hệ thống. Thứ nhất, cách phân rã các thành phần và cách thực hiện công việc của các thành phần của hệ thống như thế nào. Thứ hai, các thành phần phối hợp với nhau trong quá trình thực hiện mục tiêu tổng quát của hệ thống bằng cách nào.

Mục đích của chúng ta trong bài báo này tập trung xem xét vấn đề thứ hai tức là cách mà các thành phần của hệ thống thực hiện được mục tiêu tổng quát. Để thực hiện được nhiệm vụ chung, các thành phần sẽ phải tương tác lẫn nhau hoặc theo kiểu phối hợp hành động hoặc theo kiểu cạnh tranh nhằm chia sẻ tài nguyên như thông tin... Từ quan điểm cấu trúc xã hội [3], tương tác trong hệ thống phần mềm cũng giống như các mối quan hệ xã hội trong thế giới con người, rất đa dạng và phức tạp. Để hiểu rõ hơn tương tác cần thiết cho các hệ thống phức tạp, chúng ta sẽ phân tích các đặc trưng của tương tác trong hệ hướng đối tượng quen thuộc.

2.1. Tương tác trong hệ hướng đối tượng

Các đối tượng trong hệ hướng đối tượng là các thành phần tĩnh và tương tác giữa chúng chỉ đơn giản là sự trao đổi thông điệp qua lại giữa các bên. Thông điệp được truyền bằng cách gọi thực hiện một phương thức nào đó ở đối tượng bên kia. Đối tượng nhận sẽ thực hiện hành động tương ứng theo yêu cầu. Chúng ta sẽ dễ cảm nhận được mối tương tác giữa các đối tượng là quan hệ một chiều. Đối tượng nhận thông điệp chỉ việc thực hiện yêu cầu và thông báo lại kết quả (nếu thành công) hoặc lỗi (khi không thành công). Một điểm nữa trong tương tác hướng đối tượng là sự kích hoạt lẫn nhau không có sự lan tỏa, tức là khi một đối tượng nhận được yêu cầu, nó sẽ thực hiện yêu cầu này chỉ trong khả năng của mình mà không hề có sự liên hệ với các đối tượng khác để tìm kiếm thêm thông tin cũng như nhờ sự hỗ trợ hay tương trợ. Chính điểm thiếu lan tỏa này làm cho tương tác giữa các đối tượng kém hiệu quả. Chúng ta sẽ phân tích cẩn thận hơn tương tác trong hệ hướng đối tượng để hiểu được hạn chế của cách tiếp cận này.

¹ Để tiện lợi cho trình bày, trong bài báo này, chúng tôi tạm thời sử dụng thuật ngữ “role” và “vai trò” đều có nghĩa như nhau.

Trước hết, ta thấy mặc dù đối tượng đóng gói trong nó cả trạng thái và hành vi nhưng thực chất nó hoàn toàn bị động, nghĩa là nó phải nhận được một sự kích hoạt nào đó trước khi thực sự thực hiện một hành động. Nói cách khác, đối tượng *Obj1* không có được khả năng tự thân kích hoạt hành vi của mình mà phải nhờ vào một đối tượng khác. Để thực hiện hành vi của mình một đối tượng nào khác *Obj2* sẽ phải gửi một lời gọi thực thi một phương thức cho *Obj1* khi phương thức này cho phép truy cập kiểu *public*. Một khi phương thức được gọi, hành động tương ứng sẽ được thực hiện. Điều này cũng đồng nghĩa với việc, đối tượng không có khả năng lựa chọn hành động mà việc thực hiện hành động là do đối tượng khác áp đặt lên.

Thứ hai, như đã trình bày ở trên, đối tượng là hoàn toàn bị động nên có thể coi đối tượng chỉ là những kẻ biết nghe lời đối tượng khác. Do vậy, phương pháp hướng đối tượng chỉ có thể áp dụng cho các ứng dụng trong môi trường mà sự cộng tác đã được xác định trước và dễ điều khiển. Nó không thích hợp với các môi trường phức tạp hoặc có tính cạnh tranh cao. Hạn chế này chính là do quan điểm đặt tất cả nhiệm vụ gọi, và quyết định thực hiện các hành vi lên khách thể còn chủ thể không có vai trò gì trong việc này. Những nghiên cứu về tổ chức xã hội và khoa học chính trị cho thấy các phương pháp một chiều như vậy thường không phát triển tốt [2]. Công việc sẽ tiến hành tốt hơn khi để cho những đối tượng thực hiện hành động có một tiếng nói nào đó, tức là việc gọi thực hiện một hành động trở thành một quá trình thỏa thuận giữa các đối tượng với nhau. Như vậy, đối tượng thực hiện hành động sẽ hiểu rõ hơn về hành động mà nó cần phải thực thi, từ đó biết được lý do mà nó được gọi thực hiện hành động cụ thể trong tình huống hiện thời. Chỉ có như vậy, đối tượng thực hiện mới có thể từ chối yêu cầu hoặc ít nhất là chỉ ra những hậu quả tiềm tàng nếu nó thực hiện hành động theo yêu cầu ấy. Những nhận định này trở nên đúng đắn hơn khi phần mềm chuyển từ hoạt động trong môi trường chịu sự chi phối của một tổ chức đơn lẻ (hoặc nhóm các tổ chức hợp tác với nhau) sang môi trường mở mà ở đó các tổ chức phải cạnh tranh với nhau.

Thứ ba, các phương pháp hướng đối tượng hỗ trợ rất ít cho cấu trúc tổ chức – nền tảng của tương tác trong các hệ thống phức tạp. Các mối quan hệ, tương tác trong hệ hướng đối tượng chỉ được định nghĩa đơn giản bằng cấu trúc lớp – kế thừa.

Đặc điểm cuối cùng là phương pháp hướng đối tượng không cung cấp một tập khái niệm và các kỹ thuật đầy đủ để mô hình hóa các kiểu hệ thống khác nhau. Đối với các hệ thống phức tạp, đối tượng, lớp và module là những phương tiện cốt yếu nhưng chưa đủ để trừu tượng hóa. Các đối tượng riêng lẻ là một tập hợp các phương thức và việc gọi thực hiện phương thức đó được cho là quá đơn giản nên không thể mô tả hết các kiểu tương tác xảy ra trong hệ thống phức tạp.

Việc nhận thức những nhược điểm của các phương pháp hướng đối tượng đã dẫn đến sự phát triển những kỹ thuật trừu tượng hóa mạnh hơn như mẫu thiết kế, khung ứng dụng... Trong khi chưa khẳng định được tiến bộ nào thì các kỹ thuật này đã gặp phải nhiều trở ngại như không thể đáp ứng được các yêu cầu của phát triển hệ thống phức tạp. Phương pháp luận hướng agent được cho là ứng cử viên nhằm khắc phục những hạn chế của phương pháp hướng đối tượng cũng như các mẫu thiết kế, khung ứng dụng... nhờ các đặc trưng thông minh, tự chủ và di động của các thành phần agent. Phần tiếp theo trình bày mô hình tương tác của các agent.

3. MÔ HÌNH TƯƠNG TÁC DỰA TRÊN ROLE TRONG HỆ ĐA AGENT

3.1 Mô hình tương tác trong hệ đa agent

Trong các hệ thống đa agent, nhiệm vụ chính của hệ thống được phân thành các nhiệm vụ nhỏ hơn và giao cho các agent thực thi. Mỗi agent khi đó sẽ đảm nhiệm một nhiệm vụ con và nó phải làm cách nào đó để hoàn thành sứ mệnh của mình. Cách thức mà các agent thực hiện nhằm đạt được mục tiêu đặt ra là tương tác với các agent khác để lấy thông tin, cộng tác, cạnh tranh...Càng ngày người ta càng nhận ra rằng ngoài các đặc trưng tự chủ, phân xạ, thì các đặc trưng tương tác mang tính xã hội càng trở thành quan trọng cho phát triển các hệ phần mềm phức tạp. Trong hệ thống phức tạp như tính toán hấu khắp và dịch vụ Web, các tương tác có thể được phân làm hai loại:

Tương tác kiểu cộng tác, hỗ trợ: Đây là những tương tác thường xảy ra giữa các agent thuộc cùng hệ thống nhưng thuộc các hệ thống con khác nhau. Chẳng hạn, một agent bán cần biết thông tin về các sản phẩm máy tính hiện có, nó sẽ yêu cầu cho các agent tìm kiếm để cung cấp các thông tin này. Khi đó, quan hệ giữa agent bán và agent tìm kiếm là quan hệ cộng tác.

Tương tác kiểu cạnh tranh: Thông thường, các agent thuộc các hệ thống khác nhau có mối tương tác mang tính cạnh tranh. Kiểu tương tác này xảy ra khi các agent mâu thuẫn lợi ích với nhau, tranh giành nhau để đạt được cái mình muốn, như khi cùng truy xuất đến các tài nguyên dùng chung hoặc quan hệ giữa hai agent mua và agent bán.

Như vậy, khi phát triển một hệ đa agent, bên cạnh những vấn đề liên quan đến tính toán số học trong bản thân của mỗi agent, chúng ta cần phải xem xét một vấn đề cũng hết sức quan trọng là tương tác giữa chúng. Hơn nữa, cần phải phân tách rạch ròi hai kiểu tương tác để giúp cho quá trình phát triển hệ thống dễ dàng hơn.

Ngoài ra, đối với tương tác trong các hệ thống phức tạp như tính toán hấu khắp, dịch vụ Web còn đặt ra một số yêu cầu phi chức năng trong việc phát triển ([1], [3]):

Tính tổng quát: Phương pháp phát triển của chúng ta phải tương đối tổng quát, nghĩa là mặc dù có liên quan đến một tình huống cụ thể nhưng vẫn có thể điều chỉnh được dễ dàng và có tính linh hoạt cao cho các tình huống khác.

Tính cục bộ: Một xu hướng mới trong phát triển hệ đa agent hiện nay là việc chấp nhận tính cục bộ trong tương tác giữa các agent. Nghĩa là agent được hiểu là bao gồm tập hợp các môi trường tương tác cục bộ trong đó tương tác xảy ra. Dựa vào vị trí hiện thời của mình, một agent có thể được định vị trong một môi trường tương tác cho trước và có quyền truy cập đến tài nguyên cục bộ và tương tác với các agent thực thi trong cùng môi trường.

Tính dùng lại được: Những người phát triển không nên tự làm từ đầu đến cuối một ứng dụng bất kỳ mà nên sử dụng lại những gì người khác đã phát triển khá tốt trước đó. Một phương pháp mô hình tương tác bất kỳ phải cho phép người phát triển sử dụng lại dễ dàng cả mã nguồn và giải pháp của ứng dụng.

Các tính năng hướng agent: Yêu cầu quan trọng nhất là tương tác trong hệ đa agent phải được mô hình theo một phương pháp hướng agent nào đó, nghĩa là tất cả các tính chất đặc trưng của agent phải được tính đến một cách kỹ lưỡng.

Tính thực tiễn: Bên cạnh những phương pháp hình thức để mô hình tương tác như các phương pháp kỹ nghệ hướng agent dựa trên UML, việc mô hình hóa tương tác agent phải được hỗ trợ và đơn giản hóa theo một cách thức cụ thể.

Có nhiều phương pháp khác nhau (xem [1], [4]) cho mô hình tương tác trong hệ đa agent như Message passing, Linda... Phương pháp Message passing xem tương tác đơn giản chỉ là sự trao đổi thông điệp giữa hai agent. Phương pháp Linda dựa chủ yếu trên mô hình cộng tác không cặp đôi và không gian dữ liệu chia sẻ. Nhìn chung, các phương pháp này mới chỉ dừng lại ở mức thay đổi các phương pháp tương tác trước đây bằng cách thêm vào một số khái niệm mới như “meeting point”, “event channel” hoặc “tuple space” cho phù hợp với agent mà chưa thực sự tính đến những đặc trưng của agent như tính xã hội, khả năng phản ứng và tự chủ. Để khắc phục nhược điểm này, người ta cũng đã đưa ra một vài phương pháp khác để mô hình tương tác agent dưới dạng các dịch vụ (service) và nhiệm vụ (task) như Agentis. Tuy nhiên, sự ghép nối này không đưa đến một cách tiếp cận tổng quát và hơn nữa, chưa có một phương pháp nào hỗ trợ được cho các pha phát triển phần mềm. Mỗi phương pháp chỉ áp dụng cho một trong các pha phân tích, thiết kế hoặc cài đặt. Điều này lại dẫn đến sự manh mún trong chính giải pháp phát triển phần mềm. Chính các hạn chế này đã dẫn đến sự phát triển mô hình tương tác dựa trên role và đó là mục đích của phần được trình bày trong mục đến.

3.2 Mô hình tương tác dựa trên role

Khái niệm role đã được sử dụng rộng rãi trong các phương pháp hướng đối tượng trong đó role được hiểu là một tập hành vi mà một đối tượng có thể áp đặt lên đối tượng khác để làm thay đổi khả năng cũng như hành vi của đối tượng đó [2]. Một số phương pháp khác thì xem role là một tập các đối tượng hoặc thực thể để nhấn mạnh sự tương đồng giữa role trong chương trình máy tính và trong thế giới thực.

Bài báo này theo quan điểm của nhóm Zambonelli et al. [2]: “*Role là tập các khả năng, hành vi mong đợi và tri thức mà agent có thể sử dụng đến khi cần*”. Tập các khả năng của một role là tập các hành động mà một agent đảm nhận role đó có thể thực hiện để hoàn thành nhiệm vụ của mình. Hành vi được mong đợi là tập các sự kiện mà agent phải quản lý nhằm thực hiện được nhiệm vụ của role đó. Do đó, một tương tác giữa các agent có thể được biểu diễn bởi cặp (hành động, sự kiện). Phương pháp tương tác dựa trên role có một số ưu điểm so với các phương pháp trước đó như sau:

Role cho phép tách quan tâm giữa những vấn đề về tính toán và vấn đề về tương tác trong quá trình phát triển ứng dụng hướng agent. Các vấn đề tính toán được gói gọn trong chính bản thân agent khi định nghĩa hành vi cơ bản của nó. Mặt khác, giao tiếp của agent với các agent khác sẽ được tích hợp trong role và các hành vi thêm khác. Người ta có thể phát triển agent và role độc lập nhau về mặt thời gian cũng như phương pháp. Điều này giúp tăng khả năng phân rã và do đó làm đơn giản hóa quá trình phát triển hệ thống.

Role là một khái niệm trừu tượng ở mức cao nên cho phép sự độc lập nhất định giữa các tình huống cụ thể, nhờ đó nâng cao được tính tổng quát của các phương pháp dựa trên role.

Role có thể được phát triển theo nhu cầu và quy tắc cục bộ. Nhờ vậy mà tính cục bộ được nâng cao trong khi phát triển các hệ thống phân tán có quy mô lớn.

Role cho phép sử dụng lại giải pháp cũng như sản phẩm phần mềm. Trên thực tế, role thường gắn liền với một ngữ cảnh hệ thống và độc lập với agent. Vì thế, người thiết kế có thể sử dụng lại những role đã được thực hiện tốt trước đó cho những ứng dụng tương tự. Chúng ta hoàn toàn có thể coi role là một dạng của mẫu thiết kế (design pattern) và khi đó tập các role có liên quan cùng với định nghĩa cách thức chúng tương tác với nhau được coi là một giải pháp cho một vấn đề cụ thể và nó có thể được sử dụng lại trong nhiều tình huống tương tự.

Hai yêu cầu khác là tính agent và khả năng sử dụng cụ thể không suy ra được trực tiếp từ role nhưng từng phương pháp cụ thể sẽ có cách để đáp ứng những yêu cầu này. Có nhiều mô hình tương tác dựa trên role và một số tiêu chuẩn đã được đưa ra để đánh giá, so sánh (xem [1]) các mô hình này như: nó có hỗ trợ nhiều pha trong quá trình phát triển hay không? Nó có cung cấp tập kí hiệu để mô tả role và các hành vi của agent hay không? Có đáp ứng được sự thay đổi môi trường thực thi hay không? Người phát triển có thể dùng bản phân tích dựa trên role của phương pháp này để thiết kế, cài đặt bằng phương pháp khác hay không? Dựa trên những tiêu chuẩn này mà người ta đã đưa ra một số nhận xét cho một số phương pháp luận khác nhau sau đây ([1], [5]):

AALADIN: Một phương pháp dựa trên ba khái niệm cơ bản là agent, role và group, trong đó group là một tập các agent được coi là thành phần thực hiện nhiệm vụ nhỏ nhất. Tức là nó là mức thấp nhất có thể hoàn thành được một nhiệm vụ của hệ thống. Tuy nhiên, trong phương pháp này role gắn chặt với khái niệm agent nên nó sẽ gây khó khăn cho những người lập trình muốn có được sự độc lập giữa role và agent.

ROPE: Role được định nghĩa là những thực thể riêng biệt và có thể được các agent đảm nhận một cách linh động. Phương pháp này tập trung nhiều vào sự cộng tác nên nó thiếu mối tương tác cạnh tranh giữa các agent.

GAIA: Mô hình hệ đa agent dưới dạng các tổ chức mà ở đó các role có thể tương tác. Role được dùng nhiều trong các pha phân tích và thiết kế. Role có bốn thuộc tính cơ bản là: trách nhiệm (responsibility), quyền hạn (permission), hoạt động (activity) và giao thức (protocol). GAIA cung cấp một tập kí hiệu dựa trên tập kí hiệu FUSION để đặc tả quyền hạn của role. Ngoài ra, GAIA cũng đưa ra được mô hình tương tác biểu diễn được sự phụ thuộc và mối quan hệ qua lại giữa các role trong hệ thống.

RoleEP (Role based Evolution Programming): Hỗ trợ nhiều cho việc phát triển những ứng dụng cộng tác trong đó các agent tương tác với nhau để đạt được mục tiêu chung. Phương pháp này dựa trên bốn khái niệm cơ bản là môi trường, đối tượng, agent và role. Tuy nhiên, phương pháp này cũng chỉ hỗ trợ cho duy nhất giai đoạn cài đặt trong quá trình phát triển.

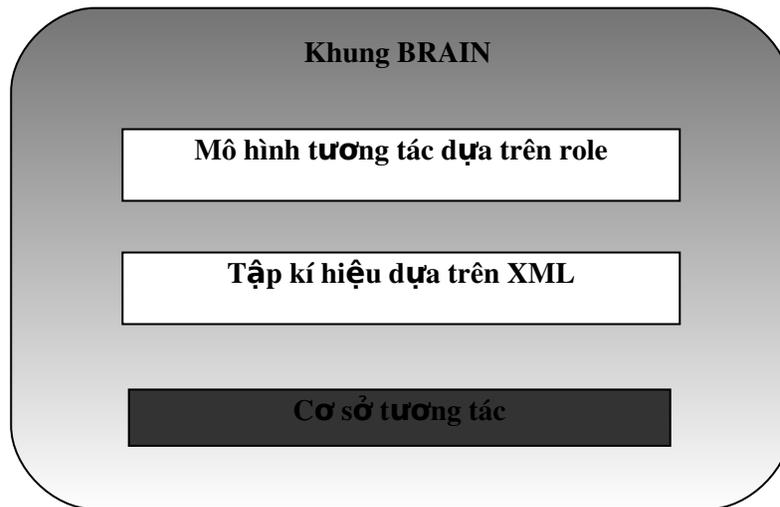
BRAIN: Hỗ trợ tất cả các giai đoạn trong quá trình phát triển hệ thống. Phương pháp này cũng có tập kí hiệu dựa trên XML để đặc tả role và hơn nữa nó còn có cả cơ sở hạ tầng tương tác (interaction infrastructure) để giúp agent quản lý các role.

Phương pháp BRAIN hội tụ nhiều ưu điểm theo quan điểm role và được chúng tôi sử dụng để mô hình hóa một dạng tương tác cụ thể là thương lượng song phương. Đó là chủ đề trong phần tiếp theo.

3.3. BRAIN Framework cho tương tác dựa trên role

Khung BRAIN (Behavioral Roles for Agent INteractions) là một khung hỗ trợ nhiều pha khác nhau trong quá trình phát triển một ứng dụng hướng agent. Khung BRAIn cung cấp 3 thành phần khác nhau, được cấu trúc thành 3 tầng khác nhau:

- i) Tầng thứ nhất: là mô hình tương tác dựa trên role
- ii) Tầng thứ hai: tập kí hiệu dựa trên XML để mô tả role
- iii) Tầng thứ ba: cơ sở tương tác có nhiệm vụ quản lý các role trong hệ thống (cơ sở này sẽ dựa trên mô hình và tập kí hiệu XML đã được định nghĩa ở hai tầng trên)

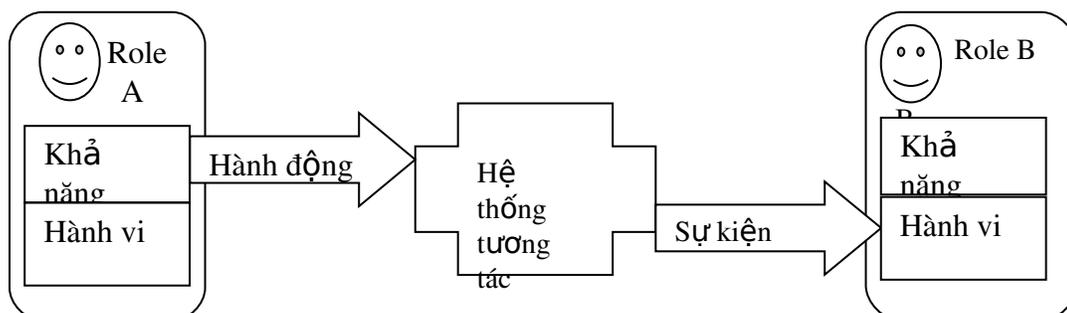


Hình . BRAIN Framework

3.3.1. Mô hình tương tác

Trong mô hình BRAIN, role được định nghĩa là một tập các khả năng và một hành vi mong đợi. Tập các khả năng là tập các hành động mà một agent đảm nhận một role có thể thực hiện để hoàn thành nhiệm vụ của nó. Một hành vi mong đợi là tập các sự kiện mà agent phải quản lý để có thể hành xử được như người dùng yêu cầu.

Tương tác giữa các agent được biểu diễn bởi một cặp (hành động, sự kiện). Cặp (hành động, sự kiện) này được hệ tương tác ở mức dưới xử lý. Hình 2 minh họa cách thức tương tác giữa hai agent. Khi agent A muốn tương tác với agent B, nó sẽ phải thực hiện một hành động (hành động này được chọn trong số các khả năng được cung cấp bởi role mà nó đảm nhận). Hành động này sẽ được dịch thành một sự kiện nhờ hệ thống tương tác ở mức dưới. Sự kiện sẽ được gửi tới agent B, ở đây B sẽ có nhiệm vụ quản lý sự kiện này một cách hợp lý.



Hình 2. Tương tác giữa hai agent trong thương lượng

Mô hình này là tổng quát, đơn giản và phù hợp với các đặc trưng của agent: Các hành động có thể được coi là một thể hiện cụ thể của tính tự chủ của agent trong khi các sự kiện phản ánh tính phản ứng của agent.

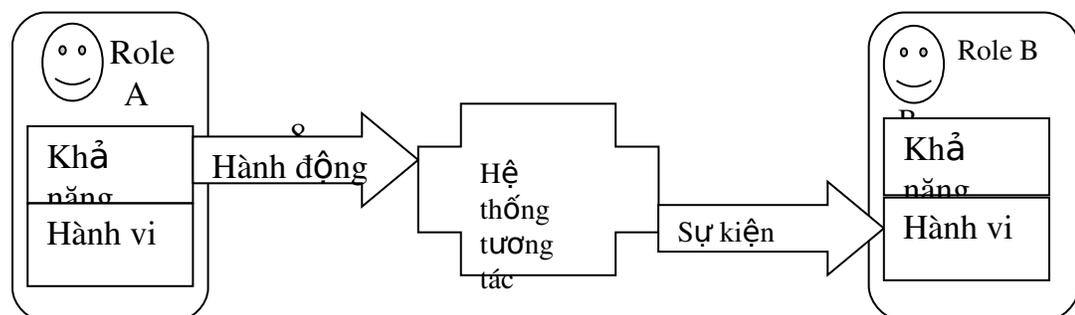
3.3.2. Tập kí hiệu

Tập kí hiệu được đề xuất trong mô hình BRAIN là XRole. Tập kí hiệu này cho phép chúng ta định nghĩa role dưới dạng XML. Việc sử dụng XML đem lại tính liên tác cao bởi nó cho phép các đối tượng phần mềm khác nhau, được phát triển bởi các công nghệ khác nhau có thể dịch và hiểu chính xác các định nghĩa role. Hơn nữa, nhờ có XML, những người phát triển phần mềm có thể có nhiều cách nhìn khác nhau về cùng một tài liệu. Điều này giúp chúng ta lựa chọn dùng chỉ những thông tin thật cần thiết. Một điểm cần lưu ý nữa là do có nhiều cách khác nhau để nhìn nhận một tài liệu nên các pha khác nhau của quy trình phát triển phần mềm có thể dùng cùng một thông tin, tạo ra sự liên tục trong suốt quá trình phát triển. Ví dụ, trong pha phân tích, phân tích viên tạo ra tài liệu Xrole theo lược đồ XML đã có trong tập kí tự Xrole của BRAIN. Chính tài liệu này sẽ chỉ ra một số tính năng của role phải định nghĩa sau này. Sau đó, những tài liệu này sẽ được dịch thành các tài liệu định dạng HTML để cung cấp bản mô tả ở mức cao tiện cho việc sử dụng ở các pha sau.

Trong pha thiết kế, các tài liệu Xrole này được dịch thành các tài liệu HTML chi tiết hơn để gọi ra những chức năng của các đối tượng liên quan. Cuối cùng, tại pha cài đặt, một lần nữa chính các tài liệu Xrole này sẽ được triển khai thành các lớp Java để thực thi các thuộc tính của role.

3.2.3. Cơ sở tương tác BRAIN

Như trong hình vẽ phía trên đã chỉ ra, BRAIN cho phép các bản cài đặt khác nhau của cơ sở tương tác có thể được gắn vào hệ thống agent một cách dễ dàng thông qua hệ quản trị platform. Cơ sở tương tác cho phép các agent đảm nhận role theo nhiều cách khác nhau. Ngoài ra, cơ sở tương tác cũng quản lý các tương tác giữa các agent với nhau theo cách nó sẽ thực hiện việc dịch một hành động thành một sự kiện và chuyển sự kiện này tới agent nhận. Cơ sở tương tác có thể điều khiển tương tác và thiết đặt các chính sách cục bộ, như cho phép hay từ chối tương tác giữa agent đảm nhận một role nào đó... Tính năng này của BRAIN tạo ra tính thích nghi cao bởi mỗi nền có thể sử dụng một nền tương tác hiệu quả nhất với mục đích của nó. Trên thực tế, không có một bản cài đặt nào là tốt nhất, việc sử dụng kiến trúc như vậy cho phép chúng ta phát triển nhiều bản cài đặt khác nhau, kiểm thử chúng để tìm ra cài phù hợp nhất với mục đích sử dụng cụ thể.



Chúng ta cũng cần phải lưu ý rằng cơ sở tương tác là tầng thấp nhất trong khung BRAIN (xem hình 1) và điều này có nghĩa là hai tầng trên sẽ không bị thay đổi theo các cơ sở tương tác. Khả năng này cho phép chúng ta sử dụng lại các phần mềm vì tất cả các tài liệu phân tích và Xrole có thể được sử dụng trong khi cài đặt cơ sở tương tác. Nói cách khác, việc sử dụng một kiến trúc 3 tầng như vậy cho phép người phát triển tập trung vào pha phân tích mà có thể bỏ qua các chi tiết về cơ sở tương tác. Điều này thực sự có ý nghĩa bởi hai agent ở hai nền với cài đặt BRAIN khác nhau vẫn có thể tương tác bằng cách trao đổi sự kiện với nhau.

Hiện nay nhóm tác giả của BRAIN đã đưa ra hai bản cài đặt khung này là RoleSystem và RoleX. Trong bài này, chúng tôi chỉ xem xét bản cài đặt RoleSystem.

Bản cài đặt RoleSystem biểu diễn role bằng các lớp trừu tượng. Trong hệ thống này, một agent muốn đảm nhận một role, nó phải đăng ký với nhân hệ thống. Khi RoleSystem nhận được yêu cầu đăng ký, nó phải kiểm tra xem agent đó có khả năng đảm nhận role yêu cầu hay không. Nếu có, nó gửi lại cho agent một thẻ xác nhận đăng ký. Thẻ này cho phép agent đảm nhận role mà nó yêu cầu, tức là có thể thực hiện được các hành động có trong role đó. Thực tế, agent thực hiện hành động và chờ sự kiện từ agent khác đều thông qua thẻ đăng ký của nó. Vậy nên có thể coi thẻ này là cầu nối giữa agent và role.

Khi agent muốn giải phóng role, agent chỉ cần giao tiếp với nhân RoleSystem để thả hồi role. Lỗi hệ thống sẽ vô hiệu hóa thẻ đăng ký của agent để nó không sử dụng thẻ này được nữa, đồng nghĩa với việc thoát khỏi môi trường tương tác.

Bản cài đặt này có ưu điểm là sử dụng và cài đặt khá đơn giản.

4. ĐẶC TẢ THƯƠNG LƯỢNG SONG PHƯƠNG DỰA TRÊN ROLE

Thương lượng là một khái niệm thường được dùng để chỉ tiến trình thỏa hiệp giữa các bên (có lợi ích xung đột nhau) để đi đến một thỏa thuận thống nhất có thể chấp nhận được. Trong thương mại điện tử ([6], [7]):

Thương lượng tự động là một quá trình thương lượng, trong đó, người mua và người bán được thay thế bằng các agent. Việc tìm kiếm, mua bán và thỏa thuận giữa người mua và người bán được thực hiện tự động bởi các agent.

Có nhiều kiểu thương lượng tự động khác nhau [7]: Thương lượng kiểu đấu giá, đấu thầu, chợ là dạng thương lượng trong mỗi phiên giao dịch có thể có một agent bán, nhiều agent mua (đấu giá), nhiều agent bán một người mua (đấu thầu) hay nhiều người bán và nhiều người mua (chợ). Loại thương lượng này liên quan nhiều đến thuộc tính giá cả của hàng hóa. Thương lượng song phương là dạng thương lượng chỉ có một bên mua và một bên bán tham gia vào quá trình giao dịch. Thương lượng có thể liên quan đến nhiều thuộc tính của hàng hóa như giá bán, chất lượng...Phần này chỉ tập trung vào xem xét đặc tả thương lượng song phương theo mô hình role.

4.1. Thương lượng song phương dựa trên ràng buộc mờ

Theo mô hình thương lượng dựa trên ràng buộc mờ có trọng số của Jennings [7], yêu cầu của bên mua được diễn đạt bằng các ràng buộc mờ đối với sản phẩm thông qua thuộc tính như *thích, rất thích...* Việc đánh giá sản phẩm mà bên bán đưa ra có thỏa mãn yêu cầu hay không cũng không thật sự chắc chắn mà phải do bên mua quyết định. Vì vậy, căn cứ vào độ ưu tiên

của các thuộc tính, bên mua cung cấp cho bên bán các ràng buộc của mình lên sản phẩm lần lượt theo thứ tự ưu tiên từ cao xuống thấp.

Đối với bên bán, họ sẽ tìm kiếm các sản phẩm thỏa mãn các ràng buộc theo thứ tự giảm dần độ ưu tiên. Giả sử, người mua cần mua một chiếc máy tính xách tay, nếu bên mua cần nhất là máy phải rẻ, sau đó là cấu hình cao... thì bên bán sẽ ưu tiên tìm sản phẩm thỏa mãn về giá cả trước, trong số sản phẩm tìm được sẽ tìm những cái thỏa mãn ràng buộc về cấu hình... Trong quá trình tìm kiếm, nếu không tìm thấy sản phẩm thỏa mãn các ràng buộc, bên bán sẽ yêu cầu bên mua nhượng bộ bớt các ràng buộc. Tuy nhiên, việc nhượng bộ không thể là vô hạn và quá trình thương lượng sẽ kết thúc khi tìm được sản phẩm thỏa mãn ràng buộc của bên mua hoặc bên mua không thể nhượng bộ hơn được nữa (chi tiết tham khảo [8], [9]).

4.2. Mô hình thương lượng song phương dựa theo role

Chúng ta xét một trường hợp cụ thể là một hệ thống mua bán máy tính nhằm minh họa cách thức hoạt động của các hệ đa agent dựa trên role. Hệ thống này sẽ phải thực hiện các công việc như sau:

Lấy thông tin yêu cầu từ người mua: giá cả sản phẩm, nhà sản xuất, màu sắc, cấu hình...

Tìm kiếm các nhà cung cấp sản phẩm.

Thương lượng để mua được sản phẩm như ý.

Như vậy, các agent đại diện người mua sẽ phải tương tác với các agent của nhiều hệ thống khác trong các môi trường khác nhau. Các chức năng của hệ thống bao gồm các role sau: Request_Getter (Lấy thông tin), Buyer (Bên mua) và Seller (Bên bán).

Vai trò Request_Getter

Role này chịu trách nhiệm lấy các yêu cầu từ người dùng. Mỗi người dùng khi có nhu cầu mua một mặt hàng nào đó sẽ phải khởi động một agent người dùng. Sau đó agent này sẽ tìm kiếm và đảm nhận một role gọi là Request_Getter. Khi đảm nhận role này, agent sẽ yêu cầu người dùng nhập vào các yêu cầu về sản phẩm cần mua. Việc lấy yêu cầu này sẽ được thực hiện thông qua role bằng cách gọi một dịch vụ trong role, ví dụ Get_Request. Sau khi lấy được yêu cầu người dùng, agent sẽ phải lưu các yêu cầu này dưới dạng sao cho dễ hiểu và chuyển được dễ dàng sang hệ thống khác. Vì vậy, role Request_Getter phải được thiết kế có một định dạng lưu trữ thông tin tốt và một phương thức để trả dữ liệu về cho agent. Định dạng tốt nhất thường được dùng để biểu diễn dữ liệu hiện nay là XML do tính khả chuyển và gọn nhẹ của nó. Để giải quyết vấn đề thứ hai là trả dữ liệu về cho agent, agent sẽ phải gọi một phương thức, chẳng hạn, store_data để lấy dữ liệu và lưu vào một biến trong. Sau khi lưu lại dữ liệu trong agent, agent người dùng sẽ giải phóng role Request_Getter và bắt đầu quá trình đi tìm các hệ thống thương mại điện tử.

Vai trò Buyer, Seller và quá trình thương lượng

Agent người sử dụng sẽ tìm kiếm người bán, đăng ký và đảm nhận role người mua Buyer. Với việc đảm nhận vai trò Buyer, nó được phép giao dịch với agent bán của hệ thống. Agent bán sẽ luôn đảm nhận vai trò Seller trong hệ thống.

Vai trò Buyer có các hành động và sự kiện sau:

Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông, Đà Lạt 2006

Hành động: send_request (gửi yêu cầu đến agent bán), make_order (đặt hàng), refuse (từ chối sản phẩm agent bán đưa ra), end (kết thúc thương lượng).

Sự kiện: list_sent (danh sách đã được gửi), downrequirement (bên bán gợi ý giảm bớt yêu cầu).

Vai trò Seller có các hành động và sự kiện sau:

Hành động: get_request (lấy yêu cầu), require_search (yêu cầu tìm kiếm thông tin), list_product (liệt kê danh sách sản phẩm đạt yêu cầu), send_list (gửi danh sách sản phẩm đến agent mua), suggest_downRequirement (đề nghị giảm bớt yêu cầu)

Sự kiện: request_sent (yêu cầu được gửi), refused (bên mua từ chối đề nghị), end (kết thúc thương lượng)

Thương lượng giữa hai agent này sẽ tiến hành theo cách thức sau:

Agent mua, qua vai trò Buyer sẽ gửi cho vai trò Seller của agent bán các thông tin về yêu cầu của người dùng.

Agent bán sẽ phân tích yêu cầu và kiểm tra xem nó có hàng hóa thỏa mãn yêu cầu không. Quá trình tìm kiếm sản phẩm đáp ứng sẽ không được trình bày ở đây.

Nếu agent bán tìm ra một danh sách các sản phẩm thỏa mãn hoàn toàn yêu cầu và gửi cho agent mua thì dựa trên quyết định của mình Agent mua sẽ tiến hành đặt. Quá trình thương lượng kết thúc tại đây.

Nếu agent bán không tìm ra sản phẩm đáp ứng mọi yêu cầu của agent mua. Nó đưa ra danh sách đáp ứng nhiều yêu cầu nhất và đề nghị agent mua giảm bớt yêu cầu chưa thỏa mãn. Nếu agent mua chấp nhận, quá trình thương lượng tiếp tục, ngược lại agent mua sẽ kết thúc thương lượng và chuyển qua hệ thống khác.

5. KẾT LUẬN

Bài báo này đã trình bày ý nghĩa tương tác giữa các thành phần trong hệ thống từ quan điểm hướng đối tượng và hướng agent. Dựa trên các phân tích, chúng tôi đã xem xét mô hình tương tác của các agent dựa trên role. Mô hình hệ đa agent dựa trên tổ chức xã hội với role đang là chủ đề thu hút nhiều quan tâm nghiên cứu. Chúng tôi cũng đã đưa ra ví dụ sử dụng mô hình dựa trên role để đặc tả dạng thương lượng song phương trong thương mại điện tử. Những trình bày trong bài báo này mới chỉ là những bước đầu nghiên cứu của chúng tôi. Nhiều vấn đề cần phải tiếp tục nghiên cứu hơn nữa như làm thế nào quản lý các role, cài đặt hệ thống dựa trên role, mô hình hình thức khái niệm role...Đó là những chủ đề nghiên cứu tiếp theo và sẽ được trình bày trong các công trình khác của chúng tôi.

TÀI LIỆU THAM KHẢO

- [1] G. Cabri, L. Ferrari and L. Leonardi, "Agent role-based collaboration and coordination: a survey about existing approaches", The 2004 IEEE Systems, Man and Cybernetics Conference, Netherlands, October 2004.
- [2] G. Cabri, L. Ferrari, L. Leonardi, F. Zambonelli, "Role-based Approaches for Engineering Interactions in Large-scale Multi-Agent Systems"
- [3] G. Cabri, L. Ferrari, L. Leonardi, F. Zambonelli, "Implementing Role-based Interaction for Internet Agents", International Symposium on applications and Internet, USA, 2003.

- [4] G. Cabri, L. Ferrari, L. Leonardi, F. Zambonelli, “BRAIN: a framework for flexible role-based interactions in multiagent systems ”, Workshop of EUMAS, UK, 2003
- [5] N. R.Jennings, M. Wooldridge, F. Zambonelli “Developing Multiagent Systems: the Gaia Methodology”, ACM Transactions on Software Engineering and Methodology”, Vol. 12, No.3, 2003, pp. 317 – 370.
- [6] M. He, N.R. Jennings and H. Liung. “On Agent- Mediated Electronic Commerce”, IEEE Transactions on Knowledge and Data Systems, 15(4), pp.985-1003, 2003.
- [7] X. Luo, N. R. Jennings et al. “A Fuzzy Constraint Based Model for Bilateral, Multi-issue Negotiations in Semi-competitive Environments”, Journal of Artificial Intelligence, V.148, No.1-2, pp.53 -102, Aug. 2003.
- [8] T.D. Que and N.M. Hung. “Agent making decision based user preferences”, Proceedings of the National Conference on Information Technology, Thainguyen, Vietnam, August 2003, pp.15-27.
- [9] T.D. Que and N.M Hung. “Bilateral negotiation among agents in e-commerce”, Proceedings of Scientific Conference of Posts and Telecommunication, Hanoi, Vietnam, Aug. 2003, pp.48-55.