

Luận văn:
“Tác tử-Công nghệ phần mềm hướng tác tử”

MỤC LỤC

MỞ ĐẦU	4
CHƯƠNG 1 GIỚI THIỆU VỀ CƠ SỞ THỰC TẬP	6
1.1 Lịch sử thành lập và phát triển của công ty qua các giai đoạn.....	6
1.2 Tổ chức bộ máy quản lý của công ty.....	7
1.3 Các sản phẩm - Dịch vụ chính của Công ty Điện toán và truyền số liệu.....	9
1.4 Định hướng phát triển.....	9
CHƯƠNG 2 CÔNG NGHỆ TÁC TỬ	12
2.1 Khái niệm về tác tử.....	12
2.2 Các đặc điểm của tác tử.....	13
2.3 Các thành phần cơ bản của tác tử.....	14
2.3.1 Kiến trúc của đơn tác tử.....	14
2.3.2 Cảm nhận và tác động.....	15
2.3.2.1 Cảm nhận.....	15
2.3.2.2 Tác động.....	17
2.3.3 Cơ chế ra quyết định.....	17
2.3.3.1 Mô hình chung.....	17
2.3.3.2 Tác tử phản xạ.....	18
2.3.3.3 Tác tử có trạng thái.....	19
2.3.3.4 Tác tử hành động có mục đích.....	21
2.3.3.5 Tác tử với cơ chế suy diễn logic.....	23
2.3.4 Hệ đa tác tử-Phối hợp trong hệ đa tác tử.....	26
2.3.4.1 Phối hợp và tầm quan trọng đối với hệ đa tác tử.....	26
2.3.4.2 Chia sẻ công việc.....	29
2.3.4.3 Chia sẻ kết quả.....	31
2.3.4.4 Phối hợp nhờ cấu trúc.....	32
2.3.4.5 Phối hợp nhờ quy tắc và luật.....	33
2.3.4.5.1 Hình thành quy tắc và luật lệ.....	33

2.3.4.5.2 Quy tắc dựng sẵn.....	35
2.3.4.6 Phối hợp thông qua ý định chung.....	36
2.3.4.7 Phối hợp nhờ lập kế hoạch.....	39
2.5 Các lĩnh vực ứng dụng.....	40
2.5.1 Ứng dụng trong quản lý sản xuất.....	40
2.5.2 Tác tử quản lý quá trình và luồng công việc(workflow).....	40
2.5.3 Tác tử thu thập và quản lý thông tin.....	41
2.5.4 Tác tử phục vụ thương mại điện tử.....	41
2.6 Ưu nhược điểm của tác tử và công nghệ tác tử.....	42
CHƯƠNG 3 CÔNG NGHỆ PHẦN MỀM HƯỚNG TÁC TỬ.....	45
3.1 Tiếp cận hướng tác tử cho công nghệ phần mềm.....	45
3.2 Phần mềm hướng tác tử là gì?.....	47
3.3 Tiếp cận hướng tác tử cho các hệ thống phần mềm.....	50
3.3.1 Các phân rã hướng tác tử.....	50
3.3.2 Các trừu tượng hoá hướng tác tử cho các hệ thống phần mềm phức tạp.....	52
3.3.3 Sự thay đổi các cấu trúc trong tổ chức tạo quản lý mềm dẻo.....	53
3.4 Vòng đời phần mềm hướng tác tử.....	54
3.4.1 Đặc tả (Specification).....	54
3.4.2 Thực hiện (Implementation).....	56
3.4.2.1 Làm mịn (Refinement).....	57
3.4.2.2 Việc thực hiện trực tiếp các đặc tả tác tử.....	57
3.4.2.3 Việc biên dịch các đặc tả tác tử.....	59
3.4.2.4 Sự xác minh.....	61
3.4.3 Các hướng tiếp cận tiêu đề (axiomatic).....	61
3.4.3.1 Sự tiên đề hoá hai ngôn ngữ đa tác tử.....	62
3.4.3.2 Các hướng tiếp cận ngữ nghĩa: kiểm tra mô hình.....	62
3.5 Phương pháp luận hướng tác tử.....	64

3.5.1 Phương pháp Prometheus.....	64
3.5.2 Phương pháp Tropos.....	65
3.5.3 Phương pháp Gaia.....	66
3.6 Một số ví dụ về ứng dụng công nghệ tác tử.....	68
Kết luận và đánh giá.....	74
Tài liệu tham khảo.....	75

MỞ ĐẦU

Ngay từ đầu những năm 80, tác tử và hệ đa tác tử (Agent và MultiAgent System) đã được biết đến với hàng loạt công trình nghiên cứu như là một hướng nghiên cứu mới. Tuy nhiên, chỉ bắt đầu từ khoảng giữa thập niên 90, tác tử và hệ đa tác tử mới được thừa nhận rộng rãi và ngay lập tức đã thu hút sự quan tâm ngày càng lớn của giới nghiên cứu cũng như giới công nghiệp trong lĩnh vực Công Nghệ Thông Tin.

Sự phát triển của kỹ thuật tính toán trong vài thập kỷ cuối đã dẫn tới những thay đổi tích cực trong các lĩnh vực sử dụng thông tin đồng thời dẫn đến sự ra đời của nhiều công nghệ và lĩnh vực nghiên cứu mới. Một mặt các hệ thống máy tính ngày càng tiên tiến cho phép xử lý thông tin nhanh hơn, đa dạng hơn đã tác động tích cực đến đời sống, văn hóa, kinh tế. Mặt khác bản thân sự phát triển và phổ cập máy tính đặt ra những yêu cầu mới về mặt công nghệ, về cách thức xây dựng, ứng dụng và nghiên cứu các hệ thống thông tin. Các hệ thống máy tính hiện đại có một số đặc điểm sau:

Việc sử dụng máy tính và thiết bị tính toán ngày càng phổ dụng. Do giá thành liên tục hạ, các hệ thống xử lý thông tin ngày càng được sử dụng nhiều trong các ứng dụng, trong các thiết bị trước đây không thể sử dụng thiết bị tính toán vì lý do kinh tế.

Máy tính ngày nay không còn là các hệ thống hoạt động riêng lẻ. Ngày càng nhiều máy tính được nối mạng cho phép liên kết, trao đổi, chia sẻ thông tin và công việc tính toán với nhau. Hệ thống thông tin dần dần có dạng các hệ thống làm việc phân tán và song song. Việc tính toán và xử lý thông tin khi đó có thể xem xét như quá trình tương tác (giữa các hệ thống tính toán). Xu hướng kết nối và xử lý phân tán được coi là đặc điểm quan trọng nhất của máy tính hiện đại.

Số lượng ứng dụng đa dạng với độ phức tạp không ngừng tăng. Máy tính ngày càng đảm nhiệm công việc phức tạp hơn, không gằn với khái niệm tính toán truyền thống. Đây là những công việc trước đây vốn chỉ có con người có khả năng

thực hiện. Nói cách khác, máy tính ngày càng trở nên “thông minh” hơn, “trí tuệ” hơn.

Máy tính ngày càng có thêm tính tự chủ. Để tăng năng suất, hiệu quả, giải phóng con người khỏi nhiều công việc truyền thống, chúng ta có xu hướng trao cho máy tính nhiều quyền hơn trong hành động và ra quyết định, đồng thời giảm bớt sự can thiệp trực tiếp của con người vào hoạt động của máy tính. Nhiều hệ thống tính toán và điều khiển có khả năng tự động hóa cao, ra quyết định độc lập làm tăng tính hiệu quả, ổn định và độ an toàn.

Các hệ thống tính toán hiện đại ngày càng có tính chất hướng người dùng. Ở các thế hệ máy tính đầu tiên, số người có thể sử dụng máy tính rất hạn chế. Họ đều là chuyên gia về máy tính hoặc lập trình viên chuyên nghiệp, được trang bị kiến thức đặc biệt để làm việc với máy tính. Ngược lại, yêu cầu với máy tính ngày nay là phục vụ người dùng càng tốt, thể hiện ở một loạt yêu cầu như giao diện thân thiện và trực giác, khả năng thích nghi với yêu cầu người dùng, cho phép cung cấp thông tin có tính cá nhân hóa với từng đối tượng sử dụng.

Để xây dựng các hệ thống tính toán thỏa mãn các đặc điểm và yêu cầu nói trên một số hướng nghiên cứu và ứng dụng mới của máy tính đã ra đời, trong đó có tác tử và hệ đa tác tử đang trở thành công nghệ của tương lai để giải quyết các vấn đề nêu trên. Vì vậy em quyết định chọn đề tài:”Tìm hiểu về công nghệ Agent” để có cái nhìn tổng quan về công nghệ tác tử, giúp người đọc hiểu qua phần nào về công nghệ Agent- công nghệ đang dần mang tính ứng dụng trong tương lai. Do thời gian có hạn và đây là một công nghệ mới nên đề tài của em không tránh khỏi những sai sót trong khi thực hiện.

Em xin chân thành cảm ơn Giáo viên hướng dẫn Nguyễn Trung Tuấn cùng toàn thể các giảng viên trong Bộ môn Công nghệ Thông Tin, các cán bộ công nhân viên trong công ty VDC1 đã giúp đỡ em trong quá trình thực tập để em có thể hoàn thành báo cáo tốt nghiệp này.

CHƯƠNG 1 GIỚI THIỆU VỀ CƠ SỞ THỰC TẬP

1.1 Lịch sử thành lập và phát triển của công ty qua các giai đoạn

- Năm 1974: Trạm máy tính của Ngành Bưu điện ra đời ở miền Bắc.

Trạm máy tính thuộc vụ Kế toán và Thống kê được thành lập theo quyết định số 539/QĐ, ngày 02 tháng 07 năm 1974, do quyền Tổng cục trưởng Tổng cục Bưu điện Vũ Văn Quý đã ký, có nhiệm vụ tính toán các số liệu theo nhiệm vụ của Vụ Kế toán và Thống kê, giúp các cơ quan, xí nghiệp thuộc Tổng cục trong công tác tính toán. Ra đời trong hoàn cảnh chiến tranh, những ngày đầu chỉ có 07 cán bộ công nhân làm việc với các máy điện cơ cá nhân của Cộng Hoà Dân Chủ Đức để thống kê số liệu cho Ngành.

- Năm 1976: Thành lập Trung tâm máy tính Ngành Bưu điện: Một trong những Trung tâm máy tính đầu tiên của cả nước

- Từ 1979 đến nay:

+ Thống nhất tổ chức máy tính toàn Ngành Bưu điện

+ Thay đổi tổ chức của Trung tâm Máy tính.

+ Trung tâm Thống kê và Tính toán Bưu điện ra đời

+ Công ty Điện toán và Truyền số liệu chính thức được thành lập ngày 26-11-1990.

+ Ngày 28 tháng 11 năm 1995, thành lập Trung tâm Điện toán và Truyền số liệu khu vực I (VDC1) có trụ sở đặt tại 75 Đinh Tiên Hoàng (HN) và Trung tâm Điện toán và Truyền số liệu khu vực III (VDC3) có trụ sở đặt tại 12 Lê Thánh Tông (Đà Nẵng).

+ Ngày 25 tháng 11 năm 1997, thành lập Trung tâm Dịch vụ Gia tăng Giá trị (VASC) trụ sở tại 258 Bà Triệu.

Hiện nay Công ty VDC đang quản lý và khai thác mạng trực Internet Việt Nam kết nối trực tiếp với xa lộ Internet quốc tế qua 3 cổng quốc gia đặt tại Hà Nội, Thành phố Hồ Chí Minh và Đà Nẵng.

1.2. Tổ chức bộ máy quản lý của công ty

1.2.1 Sơ đồ tổ chức công ty



1.2.2 Các phòng ban và chức năng

-Phòng Hành Chính:

Chức năng: Phòng Hành Chính có chức năng về công tác văn thư - lưu trữ, lễ tân, đối ngoại, thông tin tuyên truyền, nội vụ và làm đầu mối thông tin phục vụ cho hoạt động sản xuất kinh doanh của Công ty.

-Phòng Kế Hoạch:

Chức năng: Phòng Kế hoạch có chức năng về công tác Kế hoạch; Quản lý tài sản; Cung ứng vật tư.

-Phòng Kinh doanh:

Chức năng: Phòng Kinh doanh có chức năng về công tác Marketing; Kinh doanh sản phẩm, dịch vụ; Bán hàng; Hợp tác kinh doanh.

-Phòng Kỹ Thuật Điều Hành:

Chức năng: Phòng Kỹ thuật Điều hành có chức năng về kỹ thuật công nghệ, điều hành khai thác mạng và thiết bị.

-Phòng Kế Toán Tài Chính:

Chức năng: Phòng Kế toán tài chính có chức năng về công tác kế toán, thống kê, tài chính của Công ty.

-Phòng Đầu Tư Phát Triển:

Chức năng: Phòng Đầu tư Phát triển có chức năng về công tác đầu tư xây dựng cơ bản, phát triển sản xuất.

-Phòng Tổ Chức Lao Động:

Chức năng: Phòng Tổ chức Lao động có chức năng về công tác tổ chức bộ máy, nhân sự, tiền lương, đào tạo, thi đua, an ninh an toàn, chính sách đối với người lao động.

-Ban Biên Tập Báo Điện Tử:

Chức năng: Ban biên tập Báo điện tử có chức năng về thông tin quảng bá.

-Phòng Tính Cước:

Chức năng: Phòng Tính cước có chức năng về công tác tính cước và các vấn đề liên quan tới việc tính cước phí các loại hình dịch vụ trên mạng của toàn Công ty.

-Phòng Nghiên Cứu Ứng Dụng Phần Mềm:

Chức năng: Phòng Nghiên cứu ứng dụng phần mềm là bộ phận sản xuất, có chức năng về công tác nghiên cứu công nghệ tin học và sản xuất các sản phẩm tin học.

-VDCA:

Chức năng: Ban Dự án VDCA có chức năng về quản lý hoạt động, thực hiện và hỗ trợ thực hiện các dự án được Công ty giao.

-Phòng Tích Hợp và Phát Triển Hệ Thống:

Chức năng: Phòng Tích hợp và Phát triển hệ thống có chức năng chính trong công tác nghiên cứu triển khai công nghệ, tư vấn, xây dựng và phát triển các giải pháp tích hợp phục vụ hoạt động sản xuất kinh doanh và điều hành quản lý của Công ty.

-Phòng Danh Bạ:

Chức năng:

1. Thực hiện các công việc về cơ sở dữ liệu danh bạ toàn quốc trên Web;
2. Sản xuất đĩa CD-ROM danh bạ;

3. Phát triển các dịch vụ liên quan đến danh bạ;
4. Chủ động thực hiện các quan hệ hợp tác phục vụ cho các nhiệm vụ trên trên.;
5. Thực hiện các nhiệm vụ khác do Giám đốc Công ty giao.

-Phòng Quản Lý Tin Học:

Chức năng: Phòng Quản lý Tin học có chức năng về quản lý Khoa học Công nghệ và Sản xuất Kinh doanh trong lĩnh vực tin học.

-Ban Quản Lý Chất Lượng:

Chức năng: Ban Quản lý chất lượng có chức năng về công tác quản lý chất lượng trong các hoạt động của hệ thống sản xuất, kinh doanh và quản lý của Công ty.

1.3.Các sản phẩm - Dịch vụ chính của Công ty Điện toán và truyền số liệu (VDC)

VNN1260,VNN1260-

P,VNN1267,VNN1268,VNN1269,VNN999,VNN trực tiếp,Mega VNN, Wifi VNN,VPN VNN,Frame Relay,X25, Gọi 1717, FONE VNN,Email,Lưu trữ website,Thuê chỗ đặt máy chủ,Thương mại điện tử,Dịch vụ trực tuyến, Truyền báo,chế bản,Tin học, CNTT, Tư vấn, Đào tạo, Xuất nhập khẩu

1.4.Định hướng phát triển

-Công nghệ và kỹ thuật

Các sản phẩm và dịch vụ của VDC được cung cấp trên những công nghệ và phương tiện kỹ thuật tiên tiến nhất hiện nay:

- Công nghệ IP với các ứng dụng mới nhất : VPN, VoIP, FoIP (Phone-Phone, PC-PC, PC-Phone), UMS, WAP,...
- Các công nghệ truyền dữ liệu và truy nhập tốc độ cao : Frame Relay, ATM, ISDN, BISDN, xDSL,...

- Các trang thiết bị từ những nhà cung cấp hàng đầu: Sprint (Global One), Acatel, Sun Microsystems, Hewlett Packard, IBM, Compaq, Fujitsu, Cisco, Bay Network, Cabletron etc.
- Phần mềm hệ thống và quản trị mạng với UNIX (Sun Solaris, HP-UX), Microsoft Windows, SQL, HP Open View for Network Node Management Solution, Netscape Web/Mail Server, Raptor firewall etc.

Bên cạnh đó là mối quan hệ chặt chẽ với các nhà cung cấp dịch vụ hàng đầu thế giới:

- Telstra (Australia); Global One Group; Alcatel (France); Nortel (Canada); NTTCommunication, KDD, (Japan), Korea Telecom (RO Korea); Singapore Telecom; Microsoft, Oracle (USA); Hongkong Telecom (Hongkong); InfoAccess,...

Tất cả vì một mục tiêu: cung cấp những sản phẩm và dịch vụ tốt nhất cho khách hàng.

-Tầm nhìn kinh doanh

Xuất phát từ đặc điểm hoạt động trong lĩnh vực công nghệ thông tin, chất lượng dịch vụ và tốc độ là hai yếu tố quan trọng trong kinh doanh. Ngay từ khi thành lập năm 1989 chúng tôi đã không ngừng cố gắng để có thể cung cấp dịch vụ dựa trên triết lý kinh doanh "Uy Tín và Tốc độ".

Hoà cùng với xu hướng tự do hoá trong nền kinh tế cũng như trong viễn thông trên thế giới và tại Việt nam, trong các năm qua chúng tôi đã không ngừng phấn đấu thay đổi trong phương pháp và hình thức quản lý để đáp ứng được sự thay đổi, đứng vững và phát triển trên thị trường.

Năm 1999 đánh dấu việc xây dựng "*Văn hoá VDC*" với mục tiêu tạo động lực thúc đẩy tinh thần làm việc, tính tự chủ sáng tạo, tinh thần trách nhiệm cao... cho toàn bộ cán bộ nhân viên trong Công ty, cải thiện tinh thần thái độ phục vụ đối với khách hàng cũng như cải thiện các qui trình qui định kinh doanh.

-Chiến lược kinh doanh

- Luôn giữ vững và phát triển mối quan hệ với khách hàng, đối tác, bạn hàng:

Phát triển và mở rộng hệ thống hỗ trợ dịch vụ (24h/24h, 7 ngày trong tuần) thống nhất trên toàn quốc thông qua số điện thoại truy nhập 1801260, các hoạt động chăm sóc khách hàng được thực hiện trên tất cả phương tiện như điện thoại, fax, email và hỗ trợ trực tuyến thông qua Website hỗ trợ khách hàng :

- Không ngừng phát triển đa dạng hoá các dịch vụ cung cấp, tăng cường cung cấp các giải pháp tích hợp trọn gói cho khách hàng đáp ứng mọi nhu cầu, mọi khả năng chi phí, mọi nơi và mọi lúc.
- Nâng cao năng lực mạng lưới thông qua việc áp dụng các công nghệ mới, đảm bảo chất lượng dịch vụ cung cấp, xây dựng một "Hệ thống mạng khu vực" không dừng lại trong Việt nam mà mở rộng các điểm truy nhập trên thế giới tập trung vào khu vực Châu á, Mỹ.
- Hoàn thiện và phát triển "Văn hóa VDC": Xây dựng "Văn hóa VDC" là một yếu tố quan trọng dẫn đến sự thành công của Công ty trong thời gian qua, đây tiếp tục là một chiến lược quan trọng của Công ty nhằm tạo ra một phong cách làm việc mới - "*Phong cách VDC*"
- Đa dạng hóa, mở rộng quan hệ với đối tác; mở rộng phạm vi cung cấp dịch vụ: Hiện nay Công ty đã có quan hệ cung cấp dịch vụ với nhiều nhà cung cấp dịch vụ quốc tế khác nhau, phạm vi cung cấp dịch vụ hơn 150 nước trên thế giới. Trong các năm tới chúng tôi sẽ tiếp tục mở rộng quan hệ hợp tác với các đối tác trong và ngoài nước mở rộng phạm vi cung cấp dịch vụ, phấn đấu trở thành nhà cung cấp dịch vụ Viễn thông có uy tín trong khu vực Châu á - Thái Bình Dương.

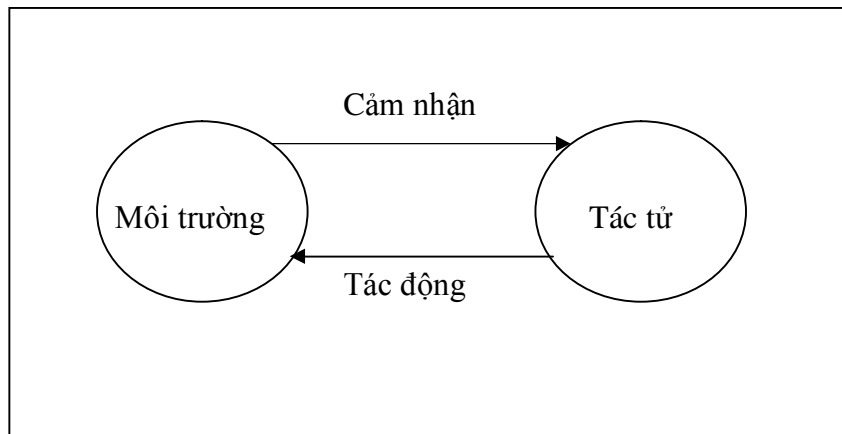
CHƯƠNG 2 CÔNG NGHỆ TÁC TỬ

2.1 Khái niệm về tác tử

Có rất nhiều định nghĩa khác nhau, thậm chí khác nhau về tác tử, nhưng có một định nghĩa thường được sử dụng như sau:

Tác tử (agent) là hệ thống tính toán hoạt động tự chủ trong một môi trường nào đó, có khả năng cảm nhận môi trường và tác động vào môi trường.

Có thể hiểu định nghĩa trên như sau: Hệ thống tính toán có thể là phần cứng, phần mềm, hoặc cả phần cứng lẫn phần mềm. Bất cứ tác tử nào cũng tồn tại và hoạt động trong một môi trường nhất định. Tác tử nhận thông tin từ môi trường qua các cơ quan cảm nhận và tác động vào môi trường bằng các cơ quan tác động.



Hình 2.1 Tác tử tương tác với môi trường

Đối với các tác tử phần cứng, cơ quan cảm nhận có thể là các cảm biến, camera, cơ quan tác động có thể là các bộ phận cơ học, quang học hoặc âm thanh. Đối với các tác tử là chương trình phần mềm, môi trường hoạt động thông thường là các máy tính hoặc mạng máy tính. Việc cảm nhận môi trường và tác động được thực hiện thông qua các lời gọi hệ thống. Nói chung, tác tử có thể được thiết kế để hoạt động để hoạt động trong nhiều dạng môi trường khác nhau. Một điểm cần chú ý là cảm nhận về môi trường của tác tử có thể không đầy đủ do môi trường quá phức tạp hoặc có chứa các yếu tố không xác định.

Một yêu cầu quan trọng đối với tác tử là tính tự chủ. Cũng như bản thân định nghĩa về tác tử, cũng có nhiều cách hiểu khác nhau về tính tự chủ. Ở đây, tự chủ được hiểu như là khả năng các tác tử hành động không cần đến sự can thiệp trực tiếp của người hay các tác tử khác: tác tử hoàn toàn có khả năng kiểm soát trạng thái cũng như hành vi của mình trong một thời gian tương đối dài. Một số các tác giả định nghĩa tính tự chủ rộng hơn, chẳng hạn yêu cầu tác tử phải có khả năng tự học.

Với đặc điểm tồn tại và hành động tự chủ trong môi trường, tác tử có thể thực hiện các mục tiêu cho trước và do vậy có thay thế chủ của mình (người dùng hoặc các tác tử khác) thực hiện một số các nhiệm vụ nào đó.

2.2 Các đặc điểm của tác tử

Một tác tử thông thường có những đặc điểm sau:

- *Tính phản xạ*: Tác tử có khả năng phản xạ kịp thời với các thay đổi trong môi trường mà tác tử cảm nhận được.
- *Tính chủ động (hành động có mục đích)*: không chỉ phản xạ, tác tử còn phải biết chủ động tìm kiếm khả năng hành động hướng tới thực hiện mục tiêu được giao.
- *Tính cộng đồng*: Tác tử có khả năng tương tác với người dùng hoặc các tác tử khác để thực hiện nhiệm vụ của riêng mình hoặc để giúp đỡ các đối tác.
- *Khả năng tự học*: Tự học hoặc học tự động là khả năng của tác tử thu thập các kiến thức mới từ kinh nghiệm thu lượm được, chẳng hạn qua các lần thành công và thất bại. Kết quả tự học phải làm cho các tác tử hành động tốt hơn, hiệu quả hơn.
- *Tính thích nghi*: Thích nghi là khả năng của tác tử tồn tại và hoạt động hiệu quả khi môi trường thay đổi. Mặc dù có nhiều nét liên quan với tính phản xạ, khả năng thích nghi của tác tử khó thực hiện và đòi hỏi nhiều thay đổi trong quá trình suy diễn của tác tử hơn. Tính thích nghi có thể thực hiện nhờ khả năng tự học từ kinh nghiệm của tác tử.

- *Khả năng di chuyển*: Là khả năng của tác tử (phần mềm) di chuyển giữa các máy tính hoặc các nút khác nhau trong mạng đồng thời giữ nguyên trạng thái và khả năng hoạt động của mình. Các tác tử có đặc điểm này được gọi là tác tử di động. Việc thiết kế và cài đặt tác tử di động đặt ra các yêu cầu đặc biệt về vấn đề an ninh hệ thống.

Có thể so sánh một tác tử có đầy đủ ba đặc điểm trên cùng với một cầu thủ đá bóng. Mục đích của cầu thủ là cùng toàn đội đưa bóng vào lưới đối phương đồng thời ngăn không cho đối phương đưa bóng vào lưới mình. Để đạt được mục đích này, cầu thủ phải tìm mọi cơ hội để đưa bóng về gần lưới đối phương và sút. Đây chính là thể hiện của tính tự chủ hành động có mục đích. Tuy nhiên, tình huống trên sân có khi cầu thủ phải thay đổi mục tiêu tạm thời, cụ thể là chuyền ngang hoặc thậm chí chuyền về. Khi đối phương vào bóng thô bạo thì mục tiêu trước mắt có thể chưa phải là sút bóng mà trước hết là giữ an toàn cho mình. Đây là thể hiện rõ ràng của tính phản xạ. Cuối cùng, cầu thủ trên sân phải có tính cộng đồng, thể hiện với việc phối hợp với đồng đội, tuân theo các chỉ dẫn của huấn luyện viên và trọng tài.

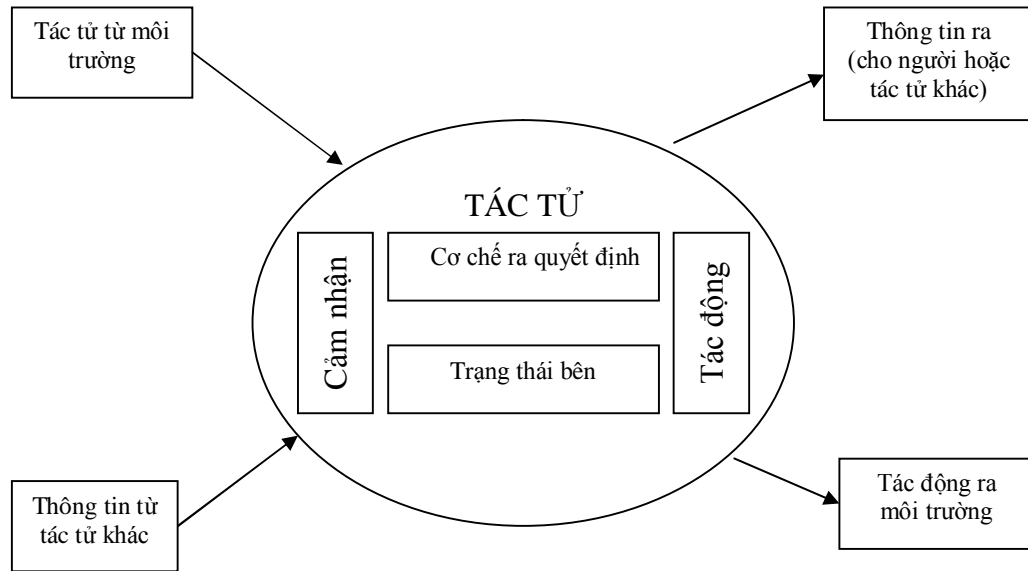
2.3 Các thành phần cơ bản của tác tử

Có thể xem xét tác tử từ hai góc độ: như một thực thể trừu tượng với cơ chế suy diễn quyết định riêng; như một thành viên trong cộng đồng tác tử với các mối quan hệ, tương tác với thành viên khác tức là ta đi xem xét tác tử như một thực thể riêng và hệ đa tác tử.

2.3.1 Kiến trúc của đơn tác tử

Ở mức độ tổng quát, tác tử có kiến trúc như hình vẽ sau:

Từ hình vẽ ta thấy, tác tử nhận thông tin từ môi trường (bao gồm thông tin từ các tác tử khác) thông qua cơ quan cảm nhận. Nhờ có cơ chế ra quyết định, tác tử lựa chọn hành động cần thực hiện. Quá trình ra quyết định có thể sử dụng thông tin về trạng thái bên trong của tác tử. Trong trường hợp đó, tác tử lưu trữ trạng thái dưới dạng những cấu trúc dữ liệu riêng. Hành động do cơ chế ra quyết định lựa chọn sau đó được tác tử thực hiện thông qua cơ quan tác động.



Hình 2.2 Kiến trúc của tác tử

Cơ chế suy diễn có thể thay đổi cho từng kiểu kiến trúc cụ thể và ảnh hưởng tới những thành phần khác. Chẳng hạn có thể có kiến trúc trong đó quá trình suy diễn không sử dụng tới trạng thái bên trong và do vậy tác tử không cần lưu giữ các thông tin này. Đối với các tác tử có thêm khả năng khác như học tự động, kiến trúc tác tử có thể có thêm thành phần riêng để thực hiện các chức năng này.

2.3.2 CẢM NHẬN VÀ TÁC ĐỘNG

2.3.2.1 CẢM NHẬN

Cơ chế cảm nhận cho phép tác tử biết được những gì đang diễn ra xung quanh, từ đó ra quyết định và hành động phù hợp. Đối với người và động vật, quá trình cảm nhận được thực hiện qua những giác quan, còn với tác tử phần cứng như robot cơ quan cảm nhận là cảm biến, camera. Tác tử phần mềm, quá trình cảm nhận có thể diễn ra một cách chủ động bằng cách trao đổi thông điệp với các tác tử khác hoặc thụ động thông qua nhận và xử lý các sự kiện hoặc thông điệp do hệ điều hành gửi tới.

Một yêu cầu quan trọng đối với cơ chế cảm nhận của tác tử là khả năng lọc những thông tin có ích và liên quan trực tiếp tới tác tử. Quá trình cảm nhận có thể mô tả một cách hình thức như sau. Gọi P là tập các hình dung của tác tử về trạng

thái môi trường. Chức năng cảm nhận của tác tử có thể coi như một hàm: *cảm_nhận*: $S \rightarrow P$ cho phép ánh xạ từ trạng thái môi trường vào hình dung của tác tử vào môi trường.

Một đặc điểm của hàm cảm nhận là hình dung về môi trường mà tác tử nhận được có thể không đầy đủ. Bên cạnh đó, nhiều trạng thái khác nhau của môi trường có thể được cảm nhận giống nhau. Giả sử $s_1 \in S$ và $s_2 \in S$ là hai trạng thái môi trường khác nhau ($s_1 \neq s_2$) sao cho $cảm_nhận(s_1) = cảm_nhận(s_2)$. Khi đó hai trạng thái môi trường khác nhau được ánh xạ thành cùng một hình dung của tác tử. Nói cách khác tác tử có cùng cảm nhận về những môi trường khác nhau. Xét một tác tử làm nhiệm vụ quản lý thư điện tử. Giả sử có hai quan sát như sau:

Quan sát x:

“có thư mới”

Quan sát y:

“máy tính có trang bị web camera”

nếu đây là hai quan sát duy nhất mà ta quan tâm thì môi trường có thể có một trong bốn trạng thái sau:

$$S = \{ \langle x, y \rangle, \langle \neg x, y \rangle, \langle x, \neg y \rangle, \langle \neg x, \neg y \rangle \} = \{ s_1, s_2, s_3, s_4 \}$$

Trong đó x có nghĩa là sự kiện x xảy ra (có thư mới) và $\neg x$ có nghĩa là sự kiện x không xảy ra (không có thư mới). Ký hiệu tương tự cho y. Như vậy trạng thái s_1 là trạng thái có thư mới và máy tính có trang bị camera, s_2 là trạng thái không có thư mới và máy tính có trang bị camera.

Giả sử lọc email chỉ quan tâm đến sự kiện có email và do vậy chỉ thực hiện được quan sát x. Các trạng thái trong đó chỉ có giá trị của y khác nhau được coi là như nhau đối với tác tử. Gọi P_1, P_2 là hai cảm nhận của tác tử cho trường hợp có thư mới và không có thư mới. Hàm cảm nhận khi đó sẽ có dạng sau:

$$cảm_nhận \begin{cases} p_1\text{neus} = s1\text{hoacs} = s3 \\ p_2\text{neus} = s2\text{hoac} = s4 \end{cases}$$

Trong trường hợp đó ta ký hiệu $S_1 = S_3$ và $S_2 = S_4$. Đối với tác tử, S_1 và S_3 là tương đương, S_2 và S_4 là tương đương. Toàn thể các trạng thái của môi trường khi đó được chia thành các nhóm trạng thái, trạng thái trong mỗi nhóm được coi là tương đương đối với tác tử. Ký hiệu số nhóm tương đương như vậy là $/\equiv/$, số trạng thái của môi trường là $/S/$. Ta có $/\equiv/=/S/$ khi tác tử có thể cảm nhận được tất cả trạng thái của môi trường. Ngược lại, nếu $/\equiv/=1$, tất cả các trạng thái của môi trường đối với tác tử là như nhau.

2.3.2.2 TÁC ĐỘNG

Tác động là quá trình ngược với cảm nhận. Tác tử tiến hành tác động vào môi trường khi cơ chế suy diễn và ra quyết định chọn được hành động cần thiết. Nói chung tác tử tác động vào môi trường thông qua cơ quan tác động (hoặc cơ quan chấp hành). Đối với tác tử phần mềm, tác động có thể được thực hiện bằng cách gửi thông điệp tới các tác tử khác, thay đổi giao diện của hệ thống (tạo ra tiếng động hoặc các dấu hiệu cảnh báo), gửi thư điện tử hoặc thực hiện một số lời gọi hệ thống (gọi một số hàm của hệ điều hành).

2.3.3 CƠ CHẾ RA QUYẾT ĐỊNH

2.3.3.1 MÔ HÌNH CHUNG

Một cách tổng quát, quá trình ra quyết định của tác tử có thể mô tả như sau. Giả sử thời gian được phân chia thành những thời điểm rời rạc t_0, t_1, \dots . Tại mỗi thời điểm, tác tử phải lựa chọn một hành động từ tập hợp hữu hạn các hành động. Nhờ cơ quan cảm nhận, tác tử thu được những cảm nhận về môi trường. Giả sử tại thời điểm t_1, t_2, \dots cảm nhận của tác tử về môi trường lần lượt là p_0, p_1, \dots với $p_i \in P$, P là tập các cảm nhận có thể có của tác tử. Tại mỗi thời điểm t_i , tất cả những gì tác tử cảm nhận về môi trường cho tới thời điểm đó là chuỗi các cảm nhận $si = \langle p_0, p_1, \dots, p_i \rangle$.

Giả sử tác tử có thể thực hiện một số hành động nhất định. Gọi tập hợp các hành động mà tác tử có thể thực hiện là $A = \{a, a', \dots\}$.

Tại mỗi thời điểm t_i , tác tử có thể lựa chọn hành động $a_i \in A$ để thực hiện. Tác tử lựa chọn hành động cụ thể tùy thuộc vào chuỗi cảm nhận s_i tại thời điểm đó. Như vậy, mỗi tác tử được đặc trưng bởi một ánh xạ từ chuỗi cảm nhận sang hành động.

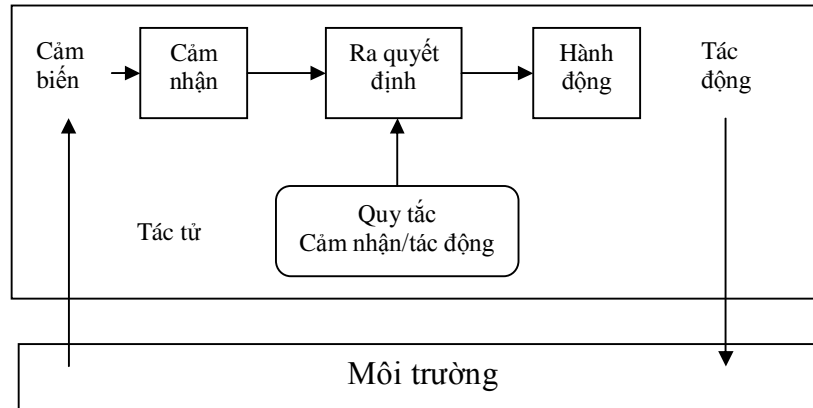
Nói cách khác, bằng cách chỉ rõ hành động mà tác tử thực hiện với chuỗi cảm nhận, ta có thể xác định cơ chế ra quyết định cho một tác tử cụ thể. Ánh xạ $S \rightarrow A$ có thể cho bằng nhiều cách. Cách đơn giản nhất là xây dựng một bảng liệt kê tất cả những chuỗi cảm nhận có thể cảm nhận và hành động tương ứng. Tuy nhiên, trong đa số trường hợp, kích thước của bảng rất lớn do số lượng chuỗi cảm nhận là vô hạn trừ khi ta hạn chế độ dài của chuỗi đó, chẳng hạn bằng cách giới hạn vòng đời của tác tử. Do vậy, phương pháp liệt kê bảng hành động là không thực tế do đòi hỏi quá nhiều bộ nhớ và thời gian tìm kiếm trong bảng.

2.3.3.2 TÁC TỬ PHẢN XẠ

Trong một số trường hợp, tác tử có thể hành động dựa trên cảm nhận hiện thời mà không cần quan tâm đến chuỗi cảm nhận trước đó. Bộ điều nhiệt đơn giản là một tác tử như vậy. Dựa trên nhiệt độ đo được tại mỗi thời điểm, trạng thái môi trường được chia thành “nóng” hoặc “bình thường”. Bộ điều nhiệt quyết định bật lò sưởi nếu nhiệt độ là “nóng” và tắt lò sưởi nếu ngược lại. Quyết định này căn cứ trên nhiệt độ hiện thời và không phụ thuộc vào những thời điểm trước đó. Hành động của tác tử là phản ứng đối với trạng thái hiện tại của môi trường. Tác tử có hành động chỉ phụ thuộc vào cảm nhận hiện thời mà không phụ thuộc vào cảm nhận trong quá khứ gọi là *tác tử phản xạ* (*reactive agent* hay *reflex agent*).

Đối với tác tử phản xạ, ánh xạ $S \rightarrow A$ trở thành ánh xạ từ tập cảm nhận vào tập hành động.

Tác tử: $P \rightarrow A$ ánh xạ này có thể xác định bởi các quy tắc có dạng “cảm nhận/hành động”, hay thường được viết dưới dạng “**nếu** cảm nhận P **thì** hành động A”. Hình 2.3 là kiến trúc của tác tử phản xạ.



Hình 2.3 Sơ đồ tác tử phản xạ

Mặc dù tác tử hoạt động hoàn toàn theo cách phản xạ được coi là tương đối đơn giản, ngay cả các tác tử phức tạp cũng có thể có một phần hành vi xác định bởi cơ chế cảm nhận/hành động như vậy. Ví dụ người và động vật đều có những phản xạ được hình thành do luyện tập hoặc là phản xạ tập trung. Các phản xạ này cho phép hành động nhanh chóng trong một số trường hợp mà không cần mất thời gian để cân nhắc kỹ càng. Khi một người bình thường chạm tay vào một vật nóng, phản xạ tự vệ thông thường là rút ngay tay lại thì vì suy nghĩ kỹ càng xem phải làm gì. Phản xạ chính là cách ra quyết định nhanh chóng như vậy.

Tác tử phản xạ đặc biệt phù hợp với những thay đổi môi trường trong đó trạng thái tại thời điểm t_i cung cấp đầy đủ thông tin cho các tác tử về toàn bộ trạng thái quá khứ trước thời điểm t_i . Môi trường như vậy được gọi là môi trường *Markov*. Môi trường *Markov* cho phép đơn giản hoá rất nhiều việc ra quyết định của tác tử.

2.3.3.3 TÁC TỬ CÓ TRẠNG THÁI

Trong nhiều trường hợp, cảm nhận hiện thời là chưa đủ để tác tử ra quyết định phải hành động như thế nào. Lấy ví dụ một tác tử lái xe. Trước khi rẽ, tác tử phải chắc chắn hai điều: thứ nhất, không rẽ vào đường cấm, và thứ hai, đường rẽ không bị xe hoặc người đi bộ cản trở. Như vậy khi quan sát không có người đi bộ (cảm nhận hiện thời) và quyết định rẽ, lái xe đã dựa vào thông tin có được từ trước (thông qua biển báo hoặc kinh nghiệm từ những lần rẽ trước) là đường rẽ không phải là đường cấm. Lý do khiến cảm nhận hiện thời là chưa đủ quyết định hành động là do tại mỗi thời điểm cơ quan cảm nhận không thể cung cấp đủ toàn bộ thông tin về trạng thái môi trường xung quanh. Để có thể hình dung được toàn thể về môi trường, tác tử phải sử dụng thông tin từ những cảm nhận trước đó. Thông tin này cho phép phân biệt những trạng môi trường khác nhau nhưng lại sinh ra cùng một cảm nhận ở thời điểm hiện tại.

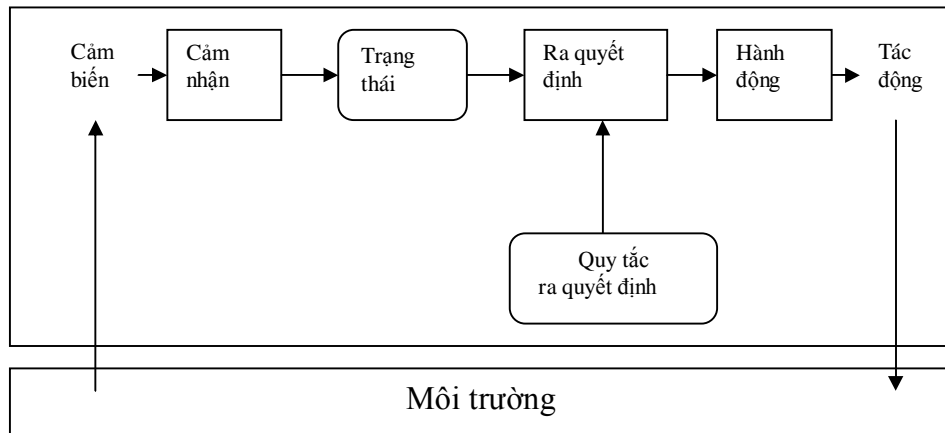
Tác tử ghi lại thông tin về môi trường bằng cách lưu lại chuỗi các cảm nhận cho tới thời điểm hiện tại. Tuy nhiên, nếu chuỗi cảm nhận dài thì việc lưu lại là không hiệu quả. Thay vào đó, tác tử duy trì một cấu trúc thông tin gọi là *trạng thái bên trong*. Trạng thái bên trong lưu trữ thông tin về môi trường mà tác tử nhận được và được cập nhật mỗi khi tác tử có cảm nhận mới. Việc lựa chọn hành động sẽ dựa trên những thông tin chứa trong trạng thái bên trong của tác tử.

Quá trình cảm nhận và ra quyết định sử dụng trạng thái bên trong diễn ra như sau: Gọi I là tập các trạng thái bên trong của tác tử, P là tập các cảm nhận. Mỗi khi tác tử có cảm nhận $p_i \in P$, cảm nhận đó sẽ làm thay đổi trạng thái bên trong của tác tử. Có thể biểu diễn sự thay đổi trạng thái theo cảm nhận qua hàm *cập nhật* như sau: *Cập nhật*: $I \times P \rightarrow I$.

Việc ra quyết định về hành động khi đó được xác định như ánh xạ từ trạng thái sang hành động. *Hành động*: $I \rightarrow A$

Sơ đồ của tác tử với trạng thái bên trong được minh họa như hình vẽ dưới đây:

Khởi đầu, trạng thái bên trong của tác tử là i_0 . Sau khi quan sát trạng thái s của môi trường, tác tử có được cảm nhận P về trạng thái đó. Trên cơ sở cảm nhận, trạng thái bên trong được thay đổi và có giá trị mới $Cập_nhật(i_0, P)$. Trạng thái mới của tác tử được ánh xạ thành hành động tiếp theo $Hành_động(Cập_nhật(i_0, P))$. Chuỗi “cảm nhận”, “cập nhật”, “hành động” tạo thành một chu kỳ hoạt động của tác tử. Chu kỳ này được lặp đi lặp lại cho đến khi tác tử chấm dứt hoạt động của mình.



Hình 2.4 Tác tử có trạng thái bên trong

2.3.3.4 TÁC TỬ HÀNH ĐỘNG CÓ MỤC ĐÍCH

Thông thường, tác tử được tạo ra để thực hiện một nhiệm vụ nào đó, và để tác tử thực hiện đúng nhiệm vụ của mình thì thông tin về trạng thái môi trường là chưa đủ. Tác tử cần có thông tin về nhiệm vụ, mục đích hoạt động của mình là gì. Và để tác tử hoạt động có mục đích là xây dựng sẵn chương trình hành động và yêu cầu tác tử hành động theo các bước ghi trong chương trình đó. Tuy nhiên, cách này có một nhược điểm là cứng nhắc theo những gì đã được sắp đặt trước vì nếu môi trường thay đổi không phù hợp với điều kiện được xây dựng trước thì tác tử sẽ không thể thích nghi với thay đổi đó. Để đảm bảo tính mềm dẻo, ta chỉ thông báo cho tác tử mục đích cần đạt được thay vì cho tác tử biết phải làm thế nào để đạt được mục đích đó.

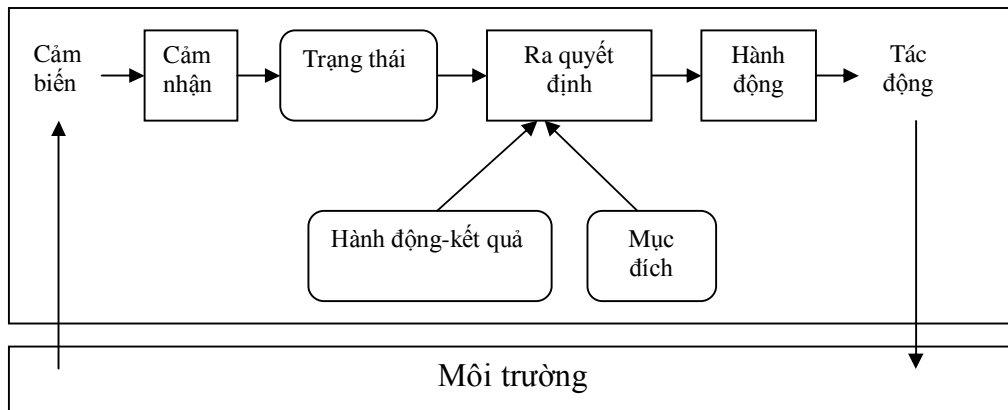
Mục đích thường có hai dạng:

- đạt được trạng thái nào đó
- đảm bảo duy trì trạng thái nào đó

Ở dạng thứ nhất, mục đích được cho dưới dạng một số trạng thái cần đạt, tác tử cần hành động sao cho môi trường chuyển sang một trong những trạng thái đó. Ví dụ, đối với tác tử làm nhiệm vụ chơi cờ vua, mục đích cần đạt được là bất cứ thế cờ nào chiếu hết đối phương.

Ở dạng thứ hai, tác tử được yêu cầu tránh một số trạng thái cho trước.

Cùng với thông tin mục đích, tác tử cần biết mỗi hành động của mình sẽ cho kết quả ra sao. Thông tin mục đích được kết hợp với thông tin về kết quả hành động để lựa chọn hành động cho phép đạt được mục đích. Việc ra quyết định trong trường hợp này khác với ra quyết định theo kiểu phản xạ ở chỗ nó đòi hỏi thông tin về tương lai: hành động nào sẽ dẫn đến kết quả nào. Trong trường hợp phản xạ, tác tử không cần thông tin loại này vì hành động cho từng tình huống cụ thể đã được tính toán trước. Quá trình ra quyết định dựa trên thông tin về mục đích đòi hỏi khối lượng tính toán lớn hơn so với phản xạ nhưng lại mềm dẻo hơn rất nhiều.



Hình 2.5 Tác tử có mục đích

Trong những trường hợp đơn giản, mục đích có thể đạt được sau một hành động duy nhất. Nhưng thường thì việc đạt được mục đích là phức tạp và đòi hỏi một chuỗi hành động.

Mặc dù có tốc độ ra quyết định chậm hơn so với phản xạ thuần túy do quá trình lập kế hoạch đòi hỏi thời gian, hành động có mục đích cho phép ra quyết định mềm dẻo hơn nhiều.

2.3.3.5 Tác tử với cơ chế suy diễn logic.

Nhiều nghiên cứu về trí tuệ nhân tạo cho rằng hành vi thông minh có thể có được bằng cách biến đổi thông tin môi trường và mục đích của tác tử về dạng biểu tượng (symbolic), sau đó thực hiện các biến đổi cần thiết trên những biểu tượng đó. Cách tiếp cận này được gọi là trí tuệ nhân tạo biểu tượng (symbolic artificial intelligence) và được sử dụng trong một số nghiên cứu để xây dựng các tác tử thông minh. Hệ thống biểu tượng thông dụng nhất để mô tả môi trường và các biểu thức logic. Phần này sẽ trình bày về tác tử với mô hình môi trường được cho dưới dạng biểu thức logic và cơ chế sử dụng phép biến đổi trên mô hình đó dưới dạng suy diễn logic.

Như với mọi hệ thống sử dụng trí tuệ nhân tạo sử dụng biểu tượng khác, để xây dựng tác tử suy diễn cần giải quyết hai vấn đề:

Biến đổi thông tin về dạng biểu tượng. Thông tin về môi trường phải được biến đổi thành mô hình biểu tượng, ở đây là biểu thức logic, cần thiết cho quá trình ra quyết định trong khoảng thời gian đủ ngắn để mô hình đó không bị cũ. Thông thường, một phần mô hình môi trường như các quy luật biến đổi và mục đích của tác tử được xây dựng từ trước, một phần khác được cập nhật trong quá trình tác tử hoạt động và cảm nhận. Lấy ví dụ một robot được trang bị camera quang học, hình ảnh mà camera thu được cần được biến đổi về dạng biểu thức logic *Phía_trước(vật cản)* cho biết phía trước đang có vật cản hay không.

Biểu diễn và suy diễn. Thông tin cần được biểu diễn phù hợp bởi biểu thức logic và có cơ chế biến đổi tự động những biểu thức này (suy diễn tự động) để sinh ra hành động. Kết quả suy diễn phải được tạo ra trong thời gian đủ ngắn để không bị lạc hậu so với sự thay đổi của môi trường.

Trên cơ sở sử dụng logic để biểu diễn và suy diễn ta có thể xây dựng tác tử với các kiểu kiến trúc và độ phức tạp khác nhau, cụ thể, có thể xây dựng tác tử phản xạ, tác tử có trạng thái và tác tử hành động có mục đích. Cơ chế suy diễn logic không hỗ trợ trực tiếp cho việc xây dựng tác tử hành động theo hàm tiện ích.

Để minh họa cho việc sử dụng logic, ta xét một mô hình tác tử suy diễn logic đơn giản sử dụng logic vị từ bậc một (first-order predicate logic). Logic vị từ bậc một là một trong những hệ thống logic truyền thống được nghiên cứu và sử dụng nhiều trong triết học, toán học và trí tuệ nhân tạo. Hệ thống logic này cho phép biểu diễn thông tin về môi trường dưới dạng các đối tượng, mỗi đối tượng có thuộc tính của mình. Giữa các đối tượng tồn tại quan hệ được biểu diễn bởi các vị từ và các hàm. Việc kết hợp đối tượng và quan hệ giữa đối tượng tạo nên sự kiện và được biểu diễn bởi câu logic.

Trạng thái bên trong của tác tử được lưu giữ dưới dạng cơ sở dữ liệu bao gồm các biểu thức logic vị từ bậc một và theo truyền thống thường được gọi là cơ sở tri thức của tác tử. Trạng thái tác tử gồm hai phần:

- Phần thứ nhất biểu diễn về những sự kiện mà các tác tử nhận được nhờ quan sát và cảm nhận môi trường. Ví dụ, phần này có thể chứa những biểu thức sau

Mở (cửa)

Vị trí (50,100)

Do cảm nhận của tác tử về môi trường không thể đầy đủ và không chính xác nên những biểu thức này có thể không phản ánh chính xác những gì đang diễn ra trong môi trường và vì vậy thường được gọi là niềm tin – tác tử “tin” rằng cửa đang mở và vị trí của mình là (50, 100) trong khi thực tế không phải như vậy.

- Phần thứ hai của trạng thái chứa các quy tắc hay các luật suy diễn logic cho phép tác tử tiến hành suy diễn và đưa ra quyết định. Ví dụ, các luật có thể có dạng

Mở (cửa) → Hành động (tiến lên)

Vị trí(1,1) → Gắn (cửa)

Một số luật cho phép suy diễn trực tiếp về hành động, một số luật khác cho phép cập nhật thông tin hoặc suy diễn về những sự kiện trong môi trường mà tác tử không cảm nhận trực tiếp được.

Gọi I là tập các trạng thái có thể có của tác tử, với mỗi trạng thái là tập các biểu thức logic như trên. Ký hiệu phân tử của I là KB_1, KB_2, \dots . Quá trình ra quyết định được quy định bởi tập các luật suy diễn chứa trong cơ sở tri thức của tác tử. Ta sử dụng ký hiệu $KB \models \phi$ nếu biểu thức ϕ có thể suy ra từ KB sử dụng sự kiện và luật chứa trong KB . Hoạt động của tác tử bao gồm cập nhật trạng thái theo cảm nhận và ra quyết định về hành động dựa theo trạng thái mới cập nhật. Hai quá trình này được thực hiện bởi hai hàm *Cập_nhật* và *Hành_động* như trình bày ở trên.

Cập_nhật: $I \times P \rightarrow I$

Hành_động: $I \rightarrow A$

Hàm *Hành_động* được định nghĩa dưới dạng thủ tục suy diễn logic tự động. Ở đây, thủ tục suy diễn có dạng đơn giản của kỹ thuật chứng minh định lý tự động (theorem proving), một kỹ thuật được nghiên cứu nhiều trong trí tuệ nhân tạo. Thủ tục suy diễn được cho dưới dạng giả mã như hình 2.6.

```
1          function Hành_động(KB:I):A
2              begin
3                  for a A do
4                      if KB /= Thực_hiện(a) then
5                          return a
6                      end-if
7                  end-for
8                  for a A do
9                      if (KB/= Thực_hiện(a)) then
10                         return a
11                     end-if
12                 end-for
13                 return null
14             end function Hành_động
```

Hình 3.5: Hàm hành động của tác tử suy diễn logic

Hình 2.6 Hàm hành động của tác tử suy diễn logic

Hàm *Hành_động* nhận tham số là trạng thái hiện thời *KB* và trả về hành động $a \in A$ hoặc null nếu không tìm được hành động thích hợp. Trước tiên, hàm *Hành_động* tìm hành động a sao cho biểu thức *Thực_hiện(a)* có thể suy ra từ sự kiện và luật suy diễn chứa trong *KB*. Đoạn (3) - (7) lần lượt chọn từng hành động a trong tập hành động có thể của mình và cố gắng chứng minh $KB \models \text{Thực_hiện}(a)$. Nếu chứng minh được, a sẽ được coi là hành động thích hợp nhất cho tình huống hiện tại, hàm ngừng thực hiện và trả về a . Rõ ràng, nếu có nhiều hành động như vậy thì hàm trả về hành động bất kỳ trong số đó.

Trong trường hợp không chứng minh được *Thực_hiện(a)*, hàm sẽ tìm một hành động nào đó không bị cấm trong tình huống hiện tại. Cách xử lý này có nghĩa như sau: nếu tác tử chưa tìm được hành động đã được xác định sẵn cho một tình huống, tác tử sẽ thực hiện một tình huống nào đó không mâu thuẫn với tình huống này với hy vọng hành động được chọn sẽ dẫn tới tình huống mới, sau đó có thể tìm hành động định sẵn cho tình huống mới. Đoạn (8) - (12) duyệt tập hành động để tìm hành động a sao cho biểu thức $\neg \text{Thực_hiện}(a)$ không thể chứng minh được từ *KB* hiện thời. Nếu không tìm được, hàm trả về giá trị null có nghĩa không hành động nào được chọn.

2.3.4 HỆ ĐA TÁC TỬ-PHỐI HỢP TRONG HỆ ĐA TÁC TỬ

Do ứng dụng ngày càng phức tạp, khả năng giải quyết vấn đề của những tác tử riêng lẻ không đáp ứng được yêu cầu đặt ra hoặc tác tử trở nên quá phức tạp. Trong trường hợp đó, hệ đa tác tử là một giải pháp thích hợp.

Hệ đa tác tử là hệ thống bao gồm nhiều tác tử có khả năng phối hợp với nhau để giải quyết được những vấn đề phức tạp mà đơn tác tử không thể giải quyết được. Sau đây ta đi tìm hiểu sự phối hợp trong hệ đa tác tử.

2.3.4.1 PHỐI HỢP VÀ TẦM QUAN TRỌNG ĐỐI VỚI HỆ ĐA TÁC TỬ

Định nghĩa phối hợp:Mục đích chính của việc xây dựng và ứng dụng hệ đa tác tử là kết hợp khả năng của nhiều tác tử tự chủ để giải quyết một số nhiệm vụ. Trong một thống có nhiều thành viên với đặc điểm tự chủ như vậy, nếu mỗi thành viên hành động theo cách của mình, không tương tác hợp lý với các thành viên khác thì toàn hệ thống sẽ hoạt động rời rạc, không ăn ý, thành viên của hệ thống, thay vì đóng góp vào giải quyết công việc chung, có thể cản trở lẫn nhau, làm ảnh hưởng đến thành viên khác. Vì vậy, khi thiết kế và xây dựng hệ đa tác tử cần đặc biệt quan tâm đến vấn đề *phối hợp (coordination)*.

Phối hợp là tổ chức và quản lý quan hệ phụ thuộc trong hành động của các tác tử sao cho toàn hệ thống hoạt động một cách thống nhất.

Cụ thể, quá trình phải cho phép đáp ứng được những yêu cầu sau:

-Đảm bảo các phần việc của nhiệm vụ chung được sắp xếp trong kế hoạch của ít nhất trong một tác tử (đảm bảo công việc sẽ được ai đó thực hiện).

-Tác tử tương tác với nhau sao cho hoạt động của tác tử được kết hợp với nhau để tạo thành một kết quả chung.

-Các yêu cầu đó phải được thực hiện trong một khoảng thời gian hữu hạn với số lượng tài nguyên tính toán hợp lý.

Sự cần thiết phải phối hợp

Sau đây là một số lý do của sự cần thiết phải phối hợp:

-*Hành động của từng tác tử phụ thuộc vào hành động của tác tử khác.* Hành động của tác tử phụ thuộc vào nhau trong hai trường hợp:

+Quyết định của tác tử này ảnh hưởng đến tác tử khác, chẳng hạn khi đá bóng việc tiền đạo chạy lên phía trước sẽ ảnh hưởng tới quyết định của tiền vệ chuyền bóng lên thay vì chuyền ngang.

+Hành động của tác tử có thể mâu thuẫn với nhau

-*Phối hợp cho phép tránh được tình trạng hỗn loạn.* Trong hệ thống bao gồm nhiều tác tử, mỗi tác tử chỉ có thể hình dung cục bộ về môi trường và hành động của mình, hành động của nhiều tác tử có thể mâu thuẫn với nhau, tình trạng hỗn loạn là rất tự nhiên và không thể tránh khỏi nếu không có cơ chế phối hợp.

-*Phối hợp cho phép đạt được những ràng buộc tổng thể.* Ràng buộc tổng thể là ràng buộc mà nhóm tác tử cần thỏa mãn trong quá trình hoạt động. Nếu mỗi tác tử làm việc riêng rẽ và đều cố gắng tối ưu hàm mục tiêu riêng của mình thì các ràng buộc này sẽ bị phá vỡ.

-*Không cá thể nào có khả năng thực hiện công việc một mình do hạn chế về tài nguyên, khả năng hoặc thông tin.* Nhiều công việc không thể hoàn thành bởi những tác tử hoạt động riêng rẽ do không đủ tài nguyên hoặc thông tin. Ví dụ không có cá nhân nào có thể xây dựng phần mềm phức tạp như hệ điều hành Windows trong một thời gian tương đối tương đối ngắn. Việc tạo ra một tác tử vạn năng bao gồm những thành phần có khả năng thực hiện có thể không thực tế hoặc kém hiệu quả, và do vậy, phối hợp nhiều tác tử có khả năng giải quyết vấn đề độc lập là giải pháp duy nhất.

Một số đặc điểm của phối hợp trong hệ tác tử

Phối hợp trong hệ đa tác tử xuất hiện trong thời gian hệ thống hoạt động. Do vậy tác tử phải có khả năng phát hiện yêu cầu phối hợp và thực hiện phối hợp như một phần trong hoạt động của mình. Đặc điểm này khác với các hệ thống phân tán truyền thống trong đó quá trình phối hợp giữa các thành phần được dự đoán trước trong quá trình thiết kế.

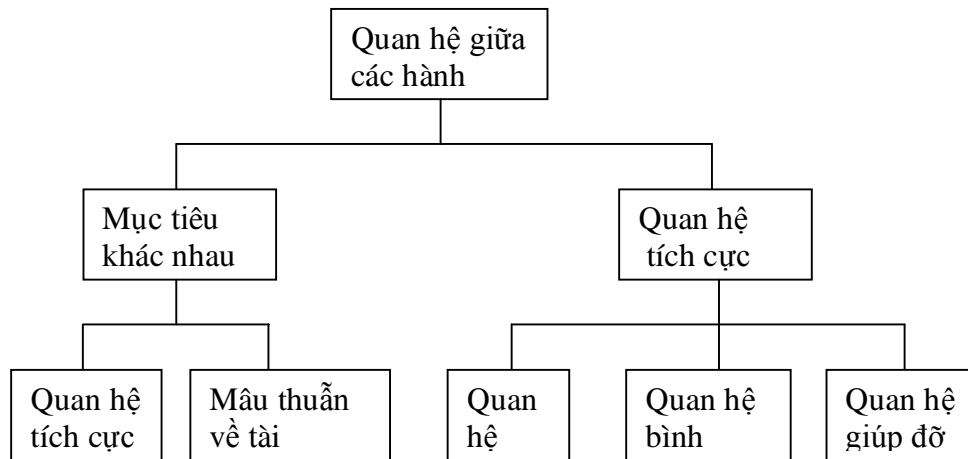
Một điều cần lưu ý là *phối hợp* và *hợp tác* không nhất thiết phải đi cùng với nhau. Nhiều tác tử hợp tác với nhau trong công việc chung không được phối hợp tốt có thể dẫn tới hỗn loạn, thiếu thống nhất. Để có thể hợp tác hiệu quả, tác tử cần lưu trữ mô hình về tác tử khác cũng như hình dung về các tác tử trong tương lai. Trong khi đó, phối hợp có thể thực hiện cho các tác tử không hợp tác với nhau. Thậm chí, điều phối có thể thực hiện đối với hệ thống bao gồm những tác tử cạnh tranh với nhau.

Để thực hiện phối hợp, tác tử có thể *liên lạc* với nhau. Tuy nhiên liên lạc không phải là điều kiện bắt buộc cho phối hợp. Phối hợp không thông qua liên lạc có thể thực hiện nếu mỗi tác tử có được mô hình của tác tử khác.

Quan hệ giữa các hành động

Như đã nói ở phần trên, hành động của tác tử trong hệ thống có thể có những mối quan hệ phụ thuộc với nhau. Khi xây dựng cơ chế phối hợp cần chú ý đến những quan hệ này. Quan hệ giữa hành động thực hiện bởi các tác tử khác nhau được chia thành hai loại chính: *tiêu cực* và *tích cực*.

Hai hoặc nhiều hành động có *quan hệ tiêu cực* hay quan hệ mâu thuẫn với nhau nếu việc thực hiện hành động này gây cản trở cho việc thực hiện đồng thời hành động khác. Nguyên nhân của quan hệ tiêu cực có thể do mâu thuẫn về mục tiêu hành động hay do hạn chế về tài nguyên. Ngược lại, *quan hệ tích cực* là quan hệ cho phép hành động này được lợi từ việc thực hiện hành động khác. Việc kết hợp hành động có quan hệ tích cực cho kết quả tốt hơn so với thực hiện các hành động một cách độc lập. Quan hệ tích cực được phân chia tiếp thành một số dạng sau:



Hình 2.7 Các dạng quan hệ giữa các hành động

-*quan hệ bình đẳng* là quan hệ khi một hành động không gắn với một tác tử cụ thể và do đó có thể được thực hiện bởi bất cứ tác tử nào.

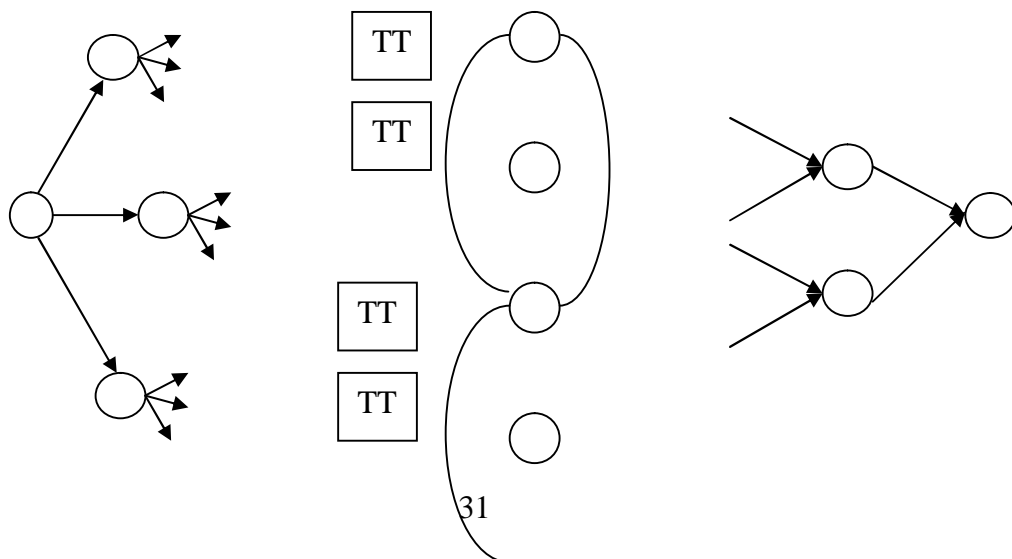
-*quan hệ gộp* là quan hệ khi hành động A của một tác tử X là một phần trong hành động B của tác tử Y, do vậy khi Y thực hiện hành động B của mình, hành động A cũng được thực hiện theo.

-*quan hệ giúp đỡ* là quan hệ trong đó việc thực hiện hành động này có lợi cho việc thực hiện hành động khác.

2.3.4.2 CHIA SẼ CÔNG VIỆC

Một trong các chiến lược thường được sử dụng để phối hợp tác tử trong việc giải quyết vấn đề phân tán là *chia sẻ công việc (task sharing)*. Ý tưởng chia sẻ công việc rất đơn giản: khi tác tử có nhiều việc phải thực hiện hoặc có những phần việc không có khả năng thực hiện, tác tử có thể yêu cầu các tác tử khác với ít việc hơn hay có khả năng hơn thực hiện một phần việc giúp mình. Chia sẻ công việc bao gồm các bước sau.

- **Phân rã công việc:** công việc được chia thành các phần việc nhỏ hơn mà các tác tử khác có thể thực hiện. Nếu cần thiết, phần việc còn lại được phân rã tiếp để tạo thành các phần việc nhỏ hơn. Quá trình này có thể tiếp tục cho đến khi từng phần việc đủ nhỏ để từng tác tử riêng lẻ có thể thực hiện được.
- **Phân phối công việc:** các phần việc được phân phối đến những tác tử thích hợp để thực hiện.
- **Thực hiện công việc:** tác tử tương ứng thực hiện phần việc được giao. Mỗi tác tử sau khi nhận phần việc của mình lại có thể tiếp tục phân chia thành những phần việc nhỏ hơn và giao cho tác tử khác thực hiện và quá trình này tiếp tục cho đến khi không cần chia sẻ tiếp.
- **Tổng hợp kết quả:** sau khi thực hiện xong phần việc của mình, tác tử gửi kết quả cho tác tử gốc (tác tử giao nhiệm vụ). Tác tử gốc kết hợp kết quả thành phần để nhận được kết quả chung.



Phân rã công việc

Phân phối và thực hiện
công việc

Tổng hợp kết quả

Hình 2.8 Mô hình chia sẻ công việc

2.3.4.3. CHIA SẺ KẾT QUẢ

Một chiến lược khác cho phép phối hợp các tác tử hợp tác là *chia sẻ kết quả*. Chia sẻ kết quả là phương pháp giải quyết vấn đề phân tán trong đó các tác tử trao đổi thông tin về kết quả thực hiện công việc của mình. Quá trình trao đổi thông tin bắt đầu từ thông tin về những kết quả đơn giản và tiếp tục bằng việc trao đổi những kết quả lớn hơn với độ phức tạp cao hơn. Một trong những lý do phải trao đổi kết quả là kết quả thực hiện công việc của từng tác tử phụ thuộc vào trạng thái và khả năng tác tử đó, do vậy kết quả thực hiện cùng một công việc trên những tác tử khác nhau có thể khác nhau. Ví dụ cùng một bài tập, những sinh viên khác nhau có thể cho lời giải khác nhau. Chia sẻ kết quả, do vậy, cho phép cải thiện kết quả chung của cả nhóm tác tử trên những phương diện sau:

- **Tăng độ tin cậy:** Kết quả sinh ra độc lập bởi các tác tử khác nhau cho cùng một công việc có thể dùng để kiểm tra lẫn nhau, củng cố ủng hộ nhau và sinh ra giải pháp tập thể có độ tin cậy cao hơn.
- **Tăng độ trọn vẹn của lời giải:** Thông thường, mỗi tác tử chỉ thực hiện một phần công việc. Chia sẻ kết quả cho phép tổng hợp kết quả các phần việc để có lời giải tổng thể và trọn vẹn cho toàn bộ công việc.
- **Tăng độ chính xác:** Để hiệu chỉnh kết quả của mình, tác tử cần biết kết quả các phần việc do tác tử khác thực hiện. Ví dụ trong trường hợp nhiều tác giả cùng tham gia viết sách, mỗi tác giả cần biết đồng nghiệp của mình viết gì để điều chỉnh nội dung viết của mình cho phù hợp.
- **Giảm thời gian thực hiện:** Ngay cả khi tác tử có thể thực hiện công việc một mình, thực hiện các phần việc song song và chia sẻ kết quả thường cho kết quả với thời gian ngắn hơn.

Mặc dù lợi ích của chia sẻ kết quả là rất rõ ràng, để có thể chia sẻ kết quả và sử dụng kết quả được chia sẻ cần giải quyết hai vấn đề sau. Thứ nhất, xử lý kết quả được chia sẻ thế nào cho có ích cho công việc của riêng mình. Thứ hai, do thông tin về kết quả có thể rất nhiều, cần trao đổi thông tin có thể lựa chọn để tránh chi phí truyền thông cao và quá tải mạng.

2.3.4.4 PHỐI HỢP NHỜ CẤU TRÚC

Cấu trúc tổ chức hay đơn giản *tổ chức* trong hệ đa tác tử bao gồm các quy định về chức năng, trách nhiệm, phạm vi quan tâm và thẩm quyền của từng tác tử trong hệ thống. Cấu trúc tổ chức gán cho mỗi tác tử một số loại công việc mà tác tử có thể thực hiện, cũng như thứ tự ưu tiên khi tác tử đồng thời nhận được nhiều công việc thuộc về những loại khác nhau. Ví dụ, trong một đội bóng nhiệm vụ của thủ môn là bảo vệ khung thành của đội mình do vậy trong đa số trường hợp thủ môn không cần chạy lên phía trước để ghi bàn.

Do mỗi tác tử có một trách nhiệm nhất định, tác tử đó cần nhận được thông tin về các kết quả thành phần có ảnh hưởng đến việc hoàn thành trách nhiệm của mình, trong khi không cần quan tâm đến những kết quả không liên quan đến việc thực hiện trách nhiệm đó.

Như vậy cấu trúc tổ chức cho phép xác định ai là người quan tâm đến một kết quả thành phần nào đó. Cấu trúc tổ chức cũng quy định mức độ hành động của tác tử khi nhận được kết quả thành phần (chẳng hạn tác tử có độ ưu tiên cao hơn có thể thay đổi kết quả của tác tử mức dưới, trong khi tác tử mức dưới không có quyền này).

Cấu trúc tổ chức có nhiều điểm tương tự tổ chức trong các cơ quan của con người đó là mỗi người có trách nhiệm thực hiện một số loại công việc, mỗi người đều biết cần gửi kết quả cho ai, nhận kết quả từ ai và có quyền phản ứng lại kết quả nhận được như thế nào.

Sau đây là một số kiểu cấu trúc tổ chức được nghiên cứu và sử dụng nhiều:

-Phân cấp: thẩm quyền ra quyết định và điều khiển được tập trung trên một tác tử hoặc nhóm tác tử ở mỗi mức của phân cấp. Liên lạc được thực hiện theo

chiều trên xuống và dưới lên giữa tác tử cấp trên và cấp dưới, tác tử cùng cấp không liên lạc trực tiếp với nhau. Tác tử mức trên thực hiện quản lý tài nguyên và ra quyết định.

-Cộng đồng chuyên gia: Đây là tổ chức cấu trúc không phân cấp. Các tác tử đều ngang hàng nhau, mỗi tác tử có một chuyên môn riêng trong lĩnh vực nhất định. Tác tử trong cộng đồng điều phối bằng cách điều chỉnh lẫn nhau sao cho đạt được kết quả chung thống nhất.

-Cộng đồng khoa học: Lấy mẫu từ mô hình hợp tác giữa các nhà khoa học. Mỗi tác tử hình thành kết quả riêng của mình. Kết quả này sau đó được công bố cho các tác tử khác để kiểm tra, tinh chỉnh hoặc bác bỏ.

-Thị trường: tổ chức kiểu này bao gồm nhiều tác tử cạnh tranh về công việc hoặc tài nguyên thông qua cơ chế ra giá và đấu thầu. Tác tử tương tác với nhau thông qua một thông số duy nhất-giá tiền- được dùng để đánh giá dịch vụ hoặc độ tiện ích. Điều phối được thực hiện thông qua điều chỉnh giá tiền tuân theo các quy luật thị trường.

2.3.4.5 PHỐI HỢP NHỜ QUY TẮC VÀ LUẬT

Quy tắc và *luật* là hai phương pháp thông dụng trong hệ đa tác tử và nó đóng vai trò quan trọng trong hoạt động của cộng đồng tác tử. Nó cho phép hạn chế hành vi của tác tử, điều hoà sự cân bằng giữa một bên là lợi ích tự do cá nhân với một bên là mục tiêu và lợi ích của cộng đồng tác tử. Việc sử dụng quy tắc và luật cũng cho phép đơn giản hoá quá trình ra quyết định của tác tử bằng cách hạn chế hướng hành động của tác tử trong những tình huống cụ thể.

Có hai cách tiếp cận chính trong sử dụng quy tắc và luật khi xây dựng cơ chế điều phối cho hệ đa tác tử:

-Quy tắc và luật dựng sẵn: Theo cách tiếp cận này, quy tắc và luật được thiết kế sẵn và được lập trình cố định trong tác tử.

-Hình thành trong quá trình hoạt động hệ thống: Quy tắc chỉ được hình thành khi tác tử tham gia vào cộng đồng tác tử cụ thể.

2.3.4.5.1 Hình thành quy tắc và luật lệ

Vậy quy tắc và luật lệ có thể hình thành thế nào trong cộng đồng tác tử? Làm thế nào tác tử có thể đi đến thoả thuận mang tính tổng thể về quy tắc chung trong khi chỉ sử dụng thông tin cục bộ mà tác tử có? Tính tổng thể biểu hiện ở chỗ tất cả tác tử đều chấp nhận và tuân theo. Đồng thời mỗi tác tử phải quyết định chấp nhận quy tắc nào dựa trên kinh nghiệm cá nhân của tác tử đó được lưu trong trạng thái tác tử. Không có sự phân cấp hay quy định sẵn về quan hệ giữa tác tử. Và để minh hoạ ta xét ví dụ sau:

“Cho một nhóm tác tử, mỗi tác tử có hai chiếc áo: một màu đỏ và một màu xanh. Các tác tử chưa từng gặp nhau trước đó và không có thông tin về nhau. Các tác tử tham gia vào một chò trôi, trò chơi kết thúc thành công nếu tất cả các tác tử mặc áo cùng một màu. Khởi đầu, mỗi tác tử mặc ngẫu nhiên một áo đỏ hoặc xanh. Trò chơi được chia làm nhiều vòng. Ở mỗi vòng, tác tử được cặp đôi với tác tử khác, các đôi được lựa chọn ngẫu nhiên. Tác tử có thể thấy tác tử cùng đôi mặc áo gì, ngoài ra không được trao đổi gì thêm. Sau khi kết thúc mỗi vòng, tác tử có thể lựa chọn giữ nguyên màu áo hay đổi sang màu khác”.

Với quy tắc chơi như trên, không tác tử nào có cái nhìn tổng thể về toàn bộ nhóm. Tác tử phải tự quyết định thay đổi hay giữ nguyên màu áo dựa vào ký ức về màu áo của các tác tử đã gặp trong những vòng trước.

Mục tiêu đặt ra đối với ví dụ này là xây dựng một hàm gọi là *hàm cập nhật chiến lược* (hàm cập nhật màu áo). Bài toán này có thể giải quyết bằng cách sử dụng các hàm cập nhật như sau:

-Hàm số đồng: Đây là hàm cập nhật đơn giản nhất. Tác tử thay đổi màu áo đang mặc sang màu còn lại nếu cho đến thời điểm đó tác tử quan sát thấy nhiều tác tử mặc màu áo này hơn nhiều màu áo hiện tại của mình. Trong trường hợp tổng quát (nhiều hơn hai màu áo), tác tử lựa chọn chiến lược (màu áo) được quan sát thấy nhiều nhất.

-Hàm số đồng với kiểu tác tử: giống như hàm số đồng ở trên, ngoài ra tác tử được chia làm hai kiểu. Khi trông thấy tác tử cùng cặp, tác tử có thể liên lạc với tác tử cùng cặp hoặc tác tử cùng kiểu. Khi liên lạc, tác tử trao đổi thông tin về ký ức

của mình. Tác tử có thể sử dụng kinh nghiệm và ký ức tác tử khác như kinh nghiệm của riêng mình.

-**Hàm số đồng có thông báo về thành công.** Tác tử liên lạc với nhau dựa trên mức độ thành công. Khi tác tử đạt tới một độ thành công nào đó với một màu áo của mình, tác tử thông báo cho các tác tử cùng cặp. Thành công ở đây nghĩa là tác tử trông thấy đối tác khác mặc áo giống mình. Như vậy, tác tử chỉ phổ biến ký ức liên quan đến thành công thay vì phổ biến toàn bộ ký ức của mình. Cách liên lạc này tránh cho tác tử trao đổi với nhau những thông tin nhiễu.

-**Hàm thưởng tích lũy cao nhất.** tác tử có thể đánh giá mỗi màu áo lựa chọn tương ứng với một khoản tiền thưởng. Hàm thưởng tích lũy cao nhất sử dụng màu áo cho giá trị tiền thưởng tích lũy cao nhất.

Tiêu chuẩn quan trọng nhất khi lựa chọn hàm cập nhật chiến lược là thời gian hội tụ: tác tử cần bao nhiêu vòng chơi để đi đến chiến lược thống nhất. Trong trường hợp việc thay đổi chiến lược là khó khăn hoặc tốn kém, chẳng hạn chiến lược là lựa chọn hệ điều hành cụ thể cho máy tính, tiêu chí khác có thể sử dụng như không thay đổi chiến lược quá thường xuyên.

2.3.4.5.2 Quy tắc dựng sẵn

Như đã nói ở trên, quy tắc dựng sẵn được xây dựng trong giai đoạn thiết kế tác tử trước khi hệ tác tử đi vào hoạt động. Quy tắc dựng sẵn được xây dựng dưới dạng các ràng buộc có dạng sau:

$$\langle E', a \rangle$$

Trong đó: $E' \subseteq E$ là tập trạng thái môi trường

$$a \subseteq A \text{ là hành động}$$

Ý nghĩa ràng buộc là khi môi trường ở trạng thái $e \in E'$ thì hành động α bị cấm thực hiện. Quy tắc và luật được quy định như tập hợp Q của các ràng buộc trên. Tác tử (chính xác hơn, kế hoạch của tác tử) được coi là hợp lệ đối với quy tắc Q nếu tác tử không có ý định thực hiện hành động bị cấm bởi ràng buộc nào đó của Q .

Ví dụ: Để minh họa cho việc sử dụng quy tắc dựng sẵn cho điều phối hệ đa tác tử, xét ví dụ sau. Cho một không gian được chia thành các ô như bàn cờ, trên đó

có các tác tử (các robot) chuyển động. Tại mỗi ô ở cùng một thời điểm chỉ được phép có mặt một tác tử. Trong trường hợp ngược lại sẽ xảy ra tai nạn. Tác tử cần thu thập và di chuyển đồ vật từ ô này sang ô khác. Yêu cầu đặt ra là xây dựng quy tắc (hay luật) sao cho mỗi tác tử có thể di chuyển mà không xảy ra tai nạn. Mặt khác luật không được quá nghiêm đến nỗi cản trở tác tử di chuyển tới điểm cần tới.

Trước tiên, ta xem xét quy tắc chỉ cho phép tác tử di chuyển theo một lộ trình duy nhất, xác định trước, hoàn toàn không thể xảy ra tai nạn. Một ví dụ của quy tắc loại đó như sau:

“Tác tử phải di chuyển với tốc độ bằng nhau và không thay đổi. Hướng chuyển động được quy định như sau. Trên dòng chẵn, tác tử phải chuyển động về bên trái, trên dòng lẻ, tác tử phải chuyển động về bên phải. Tác tử phải chuyển động lên trên khi nằm ở cột ngoài cùng bên phải. Tác tử phải chuyển động xuống dưới khi tác tử ở cột ngoài cùng bên trái của dòng chẵn hoặc cột thứ hai từ phải sang của dòng lẻ. Chuyển động của tác tử theo quy tắc này sẽ có hình dáng giống chuyển động của con rắn”.

Quy tắc có một số đặc điểm sau:

- Chuyển động tiếp theo của tác tử là hoàn toàn xác định đối với vị trí hiện tại
- Tác tử có thể di chuyển tới bất kỳ vị trí mong muốn nào.
- Để di chuyển từ vị trí hiện tại tới vị trí mong muốn, tác tử cần thực hiện tối đa $O(n^2)$ dịch chuyển, trong đó n là kích thước của bàn cờ

2.3.4.6 PHỐI HỢP THÔNG QUA Ý ĐỊNH CHUNG

Trong mô hình BDI, ta thấy trạng thái của tác tử được biểu diễn bởi niềm tin, mong muốn và ý định, trong đó ý định đóng vai trò trung tâm trong quá trình suy diễn thực tế của tác tử. Nhờ có ý định, kế hoạch hành động của tác tử có tính ổn định và ý định cũng có vai trò quan trọng trong việc phối hợp hoạt động. Chẳng hạn, nếu một sinh viên biết bạn mình đang dự định tham dự một kỳ thi tuyển sinh thì thông tin về ý định này sẽ giúp cho sinh viên đó phối hợp hành động với bạn. Thay vì rủ bạn đi chơi thường xuyên, sinh viên này sẽ không làm như vậy vì biết bạn cần tập trung thời gian thực hiện ý định. Sinh viên này cũng có thể phối hợp với

bạn tích cực hơn bằng cách cung cấp tài liệu mà anh ta biết là cần thiết cho việc ôn thi của bạn.

Trong phần này, ngoài ý định riêng, ta cần quan tâm tới *ý định chung* của cộng đồng tác tử và ảnh hưởng của ý định chung tới quá trình điều phối. Khi nhiều tác tử có ý định đạt được hành động chung nếu mỗi tác tử có ý định đạt được mục tiêu chung và tin rằng tác tử khác cũng có ý định như vậy. Câu hỏi đặt ra là: ý định chung trong trường hợp hành động tập thể khác với ý định riêng khi hành động riêng lẻ thế nào? Sự khác nhau trước tiên thể hiện ở chỗ, khi tham gia vào hoạt động tập thể, tác tử cần có trách nhiệm đối với tác tử khác trong tập thể đó.

Ý định chung hay cam kết chung như một số tài liệu sử dụng, được định nghĩa như mục tiêu chung có tính bền vững (trương đối). Để có được ý định chung, các thành viên của tập thể phải trao đổi các niềm tin và cam kết cần thiết với nhau. Giao thức trao đổi cam kết cho phép đồng bộ các tác tử sao cho các thành viên có ý định chung cần thiết về mục tiêu của cả tập thể.

Hai khái niệm liên quan đến ý định là *cam kết* (commitment) và *quy ước* (convention). Cam kết là lời hứa về việc thực hiện ý định. Tính chất quan trọng nhất của cam kết là tính bền vững. Khi một tác tử đã cam kết, tác tử đó sẽ không bỏ cam kết này cho đến khi, vì một nguyên nhân nào đó, cam kết này trở nên không cần thiết nữa. Vậy làm thế nào xác định khi nào có thể vứt bỏ một cam kết? Điều kiện có thể vứt bỏ cam kết được xác định bởi *quy ước* tương ứng. Quy ước bao gồm các luật hay quy tắc. Quy tắc này quy định rõ khi nào có thể chấm dứt cam kết. Thông thường, cam kết được chấm dứt khi: mục tiêu chung không còn tồn tại, mục tiêu chung đã đạt được hoặc mục tiêu chung không thể đạt được.

Quy ước được chia thành *quy ước (riêng)* và *quy ước xã hội*. Quy ước cho phép tác tử từ bỏ cam kết cá nhân của mình.

Quy ước xã hội quy định tác tử phải hành động thế nào đối với các tác tử khác trong tập thể khi thay đổi cam kết chung. Một mặt, khi tác tử thay đổi cam kết về ý định chung tác tử cần cho các tác tử khác trong cộng đồng biết về thay đổi đó. Mặt khác, do hành động chung cần có sự cam kết của tất cả các thành viên, sự từ bỏ

cam kết của bất cứ thành viên nào cũng ảnh hưởng tới cố gắng của tập thể và do vậy cần đánh giá lại ý định chung. Ví dụ hình 2.10 là một quy ước xã hội đơn giản thể hiện 2 yêu cầu này.

<u>Quy ước: mô hình Cohen và Levesque</u>	
Lý do đánh giá lại cam kết	
<ul style="list-style-type: none">○ Cam kết được thỏa mãn○ Cam kết không được thỏa mãn○ Động cơ thực hiện cam kết không còn	
Hành động	
R1: Nếu	Cam kết được thỏa mãn Cam kết không được thỏa mãn Động cơ thực hiện cam kết không còn
Thì	Vứt bỏ cam kết

Hình 2.9 Ví dụ mô hình quy ước

Cam kết, quy ước và phối hợp

Ý nghĩa của cam kết đối với phối hợp trong hệ đa tác tử thể hiện ở chỗ nó cho phép tác tử dự đoán được kết quả hành động của tác tử khác. Chẳng hạn, nếu hai tác tử A và B theo đuổi một tiêu chung G^{12} bao gồm hai mục tiêu con G^1 và G^2 . Tác tử A chỉ có thể yên tâm theo đuổi mục tiêu G^1 nếu biết rằng tác tử B sẽ thực hiện G^2 hoặc ít nhất sẽ thông báo cho A biết trong trường hợp định từ bỏ mục tiêu này. Trong trường hợp A không nhận được cam kết tương tự từ B, A sẽ không thực hiện G^1 vì việc thực hiện không có ý nghĩa với đối với G^{12} nếu thiếu G^2 . Việc dự đoán kết quả thực hiện công việc của tác tử khác trên cơ sở cam kết còn cho phép tác tử điều chỉnh kế hoạch của mình để tận dụng kết quả đó. Ví dụ, nếu tác tử B cam kết thực hiện công việc G^{B1} và kết quả thực hiện cho phép tác tử A thực hiện công việc G^{A1} tốt hơn, tác tử A có thể lùi việc thực hiện G^{A1} để chờ kết quả G^{B1} từ tác tử B.

Trong trường hợp hai hay nhiều tác tử cùng theo đuổi những mục tiêu phụ thuộc vào nhau và một hoặc một số tác tử từ bỏ mục tiêu của mình vì một lý do nào đó, cần có quy ước cho phép những tác tử còn lại từ bỏ cam kết của mình. Vai trò của quy ước ở đây đảm bảo cho việc phối hợp trên cơ sở cam kết có được sự mềm dẻo cần thiết.

<u>Quy ước xã hội cơ bản</u>	
Sử dụng khi:	
	Cam kết đối với mục tiêu chung thay đổi
	Cam kết đối với hành động chung thay đổi
	Cam kết chung của một thành viên thay đổi
Hành động:	
R1:	Nếu cam kết đối với mục tiêu chung thay đổi hoặc cam kết đối với hành động chung thay đổi
	Thì thông báo cho tất cả thành viên về thay đổi
R2:	Nếu cam kết chung của một thành viên thay đổi
	Thì xác định xem cam kết chung còn giá trị không

Hình 2.10 Một ví dụ quy ước xã hội

2.3.4.7 PHỐI HỢP NHỜ LẬP KẾ HOẠCH

Tác tử có thể thống nhất hành động bằng cách lập kế hoạch sao cho từng tác tử thống nhất với kế hoạch của tác tử khác. Lập kế hoạch là xác định các hành động cần thực hiện cũng như thứ tự thực hiện hành động. Đối với tác tử hoạt động một mình, kế hoạch được lập dựa trên trạng thái môi trường cộng với mục tiêu, khả năng của riêng tác tử đó. Tuy nhiên trong trường hợp đa tác tử, kế hoạch phải tính đến ảnh hưởng của tác tử khác cũng như những ràng buộc và nghĩa vụ mà tác tử phải thực hiện khi cam kết làm việc tập thể. Lập kế hoạch có thể coi như một trường hợp đặc biệt của giải quyết vấn đề phân tán- vấn đề cần giải quyết là xây dựng một

kế hoạch. Tuy nhiên lập kế hoạch có nhiều đặc thù riêng và do vậy thường được nghiên cứu như một bài toán độc lập.

2.5 CÁC LĨNH VỰC ỨNG DỤNG

Có rất nhiều ứng dụng thành công của tác tử và hệ đa tác tử trong thực tế. Các ứng dụng này phong phú về số lượng đồng thời đa dạng về chủng loại. Dưới đây, chúng ta sẽ xem xét một số ứng dụng. Những ứng dụng được xem xét là những ứng dụng tiêu biểu thể hiện được ưu điểm của giải pháp sử dụng tác tử từ những miền ứng dụng khác nhau.

2.5.1 Ứng dụng trong quản lý sản xuất

Một ứng dụng tiêu biểu sử dụng trong quản lý sản xuất là hệ thống YAMS. Bài toán sản xuất mô tả như sau: Xưởng sản xuất hình thành từ các phân xưởng. Mỗi phân xưởng bao gồm nhiều vị trí làm việc (cỗ máy, từng công nhân riêng lẻ). Các vị trí làm việc trong từng phân xưởng tạo thành hệ thống sản xuất mềm cho phép thực hiện chọn vẹn từng công đoạn như sơn, đập, lắp ráp....Nhiệm vụ của YAMS là quản lý hiệu quả quá trình sản xuất tại những xí nghiệp như vậy. Để đồng bộ và quản lý quá trình phức tạp như vậy, YAMS được xây dựng dưới dạng các tác tử cộng tác. Mỗi xưởng và thành phần của xưởng được biểu diễn bởi một tác tử. Từng tác tử có kế hoạch và khả năng của mình. Tác tử ở mức trên, công việc được phân chia xuống tác tử phân xưởng và xuống từng vị trí công việc tùy thuộc vào khả năng. Kết quả thực hiện được báo cáo cho tác tử mức trên để giám sát và có điều chỉnh nếu cần.

2.5.2 Tác tử quản lý quá trình và luồng công việc(workflow)

Nhiệm vụ của hệ thống quá trình và luồng công việc là đảm bảo các phần việc khác nhau của một quá trình được thực hiện bởi những nhân viên hoặc bộ phận thích hợp vào thời gian cần thiết. Quá trình ở đây có thể là quá trình xử lý hồ sơ, quá trình cung tiêu nguyên liệu và sản phẩm. Trong giải pháp sử dụng tác tử, tác tử thương lượng với nhau về công việc, yêu cầu và điều kiện thực hiện. Ví dụ tiêu biểu

của hệ thống thuộc loại này là ADEPT. ADEPT là hệ đa tác tử được xây dựng cho british Telecom (BT). Nhiệm vụ của hệ thống như sau. Mỗi khi khách hàng yêu cầu cung cấp dịch vụ viễn thông, một loạt bộ phận của hãng phải tham gia đánh giá để đưa ra mức giá cho dịch vụ được yêu cầu. ADEPT mô hình hoá các bộ phận và nhân viên như những tác tử cung cấp dịch vụ đánh giá. Để giải quyết công việc của mình, tác tử thương lượng với tác tử khác để nhận được dịch vụ (đánh giá) do các tác tử này cung cấp. Kết quả thương lượng bao gồm thời gian thực hiện, chất lượng dịch vụ và giá thành của dịch vụ.

2.5.3 Tác tử thu thập và quản lý thông tin

Tác tử thuộc loại này được sử dụng chủ yếu để giải quyết vấn đề dư thừa thông tin, đặc biệt là thông tin trên internet. Do lượng thông tin trên internet quá lớn và ngày càng tăng lên, người sử dụng thông tin gặp khó khăn trong hai vấn đề sau:

-*Lọc thông tin*: Hàng ngày mỗi người phải nhận và xử lý lượng lớn thông tin dưới dạng thư điện tử, tin nhắn... trong đó chỉ có một phần nội dung đáng quan tâm. Việc lọc ra thông tin cần thiết đòi hỏi thời gian và công sức.

-*Thu thập thông tin*: Khi có nhu cầu thông tin cụ thể, người sử dụng khó tìm được thông tin mình cần do lượng thông tin trên mạng quá lớn.

Tác tử thông tin là những tác tử được xây dựng để giải quyết hai vấn đề này.

Tác tử thông tin cá nhân MAXIMS là tác tử có nhiệm vụ lọc và phân loại thư điện tử. Tác tử này có khả năng thực hiện các thao tác dán giá trị ưu tiên, xóa, chuyển tiếp thư, sắp xếp và lưu trữ thư điện tử. Điểm đặc biệt là MAXIMS có khả năng tự học cách xử lý thư bằng cách quan sát người sử dụng. Trong giai đoạn đầu, tác tử ghi lại hành động của người sử dụng và thông tin về thư như người gửi, người nhận, tiêu đề thư và từ khóa. Những thông tin này sau đó được sử dụng để dự đoán hành động khi có thư mới tới. Nếu hành động được dự đoán với độ chính xác cao thì tác tử sẽ gợi ý cho người sử dụng hoặc tự động thi hành luôn.

2.5.4 Tác tử phục vụ thương mại điện tử

Quá trình giao dịch thương mại thường bao gồm một số khâu nhất định. Để giảm thời gian, tăng hiệu quả thương mại, một số khâu có thể tự động hoá bằng

cách giao quyền ra quyết định cho tác tử. Tác tử thương mại điện tử hiện nay có thể chia thành một số dạng như:

-*Tác tử mua hàng bằng cách so sánh (comparison shopping agent)*. Đây là dạng tác tử thương mại điện tử đơn giản nhất. Tác tử loại này nhận yêu cầu mua một hàng hoá cụ thể nào đó từ người sử dụng, sau đó tự động thu thập thông tin từ các địa chỉ mua hàng có lợi nhất. Tác tử loại này hoạt động hiệu quả đối với những mặt hàng tiêu chuẩn như đồ điện tử, đĩa CD, sách, trong đó tiêu chuẩn cần quan tâm nhất là giá cả.

-*Chợ điện tử và tác tử đấu giá*. Trong hệ thống loại này, tác tử đại diện cho người bán và người mua. Tác tử tự động tìm ra đối tác thích hợp, sau đó thoả thuận điều kiện mua bán hàng hoá thay cho chủ của mình.

2.6 ƯU NHƯỢC ĐIỂM CỦA TÁC TỬ VÀ CÔNG NGHỆ TÁC TỬ

Cũng như nói đến bất kỳ một công nghệ nào khác, những câu hỏi được đặt ra là vậy công nghệ đó có ưu điểm gì so với những công nghệ khác, tại sao phải sử dụng công nghệ, khi nào nên và khi nào không nên dùng? Thông thường, một công nghệ mới được coi là hữu dụng nếu công nghệ đó đáp ứng được ít nhất một trong hai tiêu chí sau:

- Cho phép giải quyết những vấn đề mà những công nghệ đang tồn tại không giải quyết được hoặc cho những giải pháp quá tốn kém.

- Cung cấp giải pháp hiệu quả hơn đối với những vấn đề đã được giải quyết bởi những công nghệ khác.

Hệ thống thông tin có thể có mức độ phức tạp khác nhau. Từ quan điểm thiết kế và cài đặt, các hệ thống kiểu chức năng là trường hợp đơn giản nhất. Tuy nhiên, rất nhiều hệ thống thông tin hiện đại không thuộc loại này và có độ phức tạp cao hơn nhiều. Sự phức tạp ở đây là kết quả một số yếu tố:

-Tính mở của hệ thống

-Cấu trúc phức tạp

-Yêu cầu hướng người dùng

Trong các hệ thống có tính mở, thành phần và cấu trúc hệ thống không được biết trước và có thể thay đổi trong quá trình hoạt động. Ví dụ của hệ thống loại này là internet. Hệ thống có cấu trúc phức tạp là hệ thống bao gồm nhiều thành phần với nhiều kiểu quan hệ khác nhau giữa các thành phần, đòi hỏi phải có cách thiết kế và cài đặt cho phép giám độ phức tạp đó. Hướng người dùng là yêu cầu hệ thống thích nghi với người sử dụng, phản xạ kịp thời cũng như chủ động thỏa mãn yêu cầu cá nhân của từng đối tượng sử dụng.

Tác tử và hệ đa tác tử là giải pháp phù hợp cho hệ thống với những đặc điểm nói trên. Cụ thể, tác tử và hệ đa tác tử có thể cho giải pháp đơn giản, hiệu quả trong những trường hợp sau:

-Hệ thống có cấu trúc phức tạp, có thể phân tích thành những thành phần tự chủ hoặc bán tự chủ tương tác với nhau.

-Dữ liệu, thông tin, tri thức có tính phân tán và phí tổn để chuyển đổi thành dạng tập trung tương đối lớn hoặc việc chuyển đổi rất khó khăn.

-Việc thiết kế hoặc xử lý tập trung làm giảm độ tin cậy hoặc tính ổn định của hệ thống.

-Yêu cầu sử dụng lại và tích hợp vào hệ thống những thành phần mềm đã có sẵn, có thể không tương thích với nhau.

Tuy nhiên, cần xem xét kỹ việc lựa chọn tác tử kể cả cho trường hợp thỏa mãn những điều kiện này. Đa số các ứng dụng có sử dụng tác tử hiện nay đều có thể được xây dựng với các công nghệ khác. Việc một ứng dụng có bốn đặc điểm như vừa nêu trên không có nghĩa tác tử là giải pháp đầu tiên và duy nhất cho ứng dụng đó. Giải pháp tác tử chỉ nên xem xét như một trong những giải pháp có thể sử dụng trong những trường hợp như vậy. Bên cạnh đó, khi lựa chọn giải pháp sử dụng tác tử cần chú ý tới những nhược điểm của giải pháp này. Cụ thể tác tử và hệ đa tác tử không thích hợp khi ứng dụng trong các trường hợp sau:

-Thứ nhất, trong hệ thống thời gian thực hoặc hệ thống trong đó thời gian đáp ứng có ý nghĩa quan trọng. Nói chung, thời gian đáp ứng của hệ tác tử là rất khó xác định.

-Thứ hai, trong những trường hợp cần có sự điều khiển tập trung và thống nhất, cũng như yêu cầu thoả mãn những ràng buộc chung và tìm giải pháp tối ưu tổng thể. Do từng tác tử chỉ có cái nhìn không toàn diện về vấn đề cần giải quyết hoặc một phần tri thức, giải pháp của hệ tác tử thường chỉ là giải pháp tối ưu cục bộ.

-Thứ ba, trong những trường hợp yêu cầu mức độ trách nhiệm cao đối với quyết định và hành động. Sử dụng tác tử trong trường hợp này đòi hỏi người (hoặc tổ chức) có độ tin tưởng cao đối với tác tử và dám uỷ thác cho tác tử hành động tự chủ. Để có mức độ tin tưởng cần thiết, người sử dụng cần có thời gian theo dõi và làm việc với tác tử để đảm bảo tác tử không vượt quá giới hạn quyền hạn của mình.

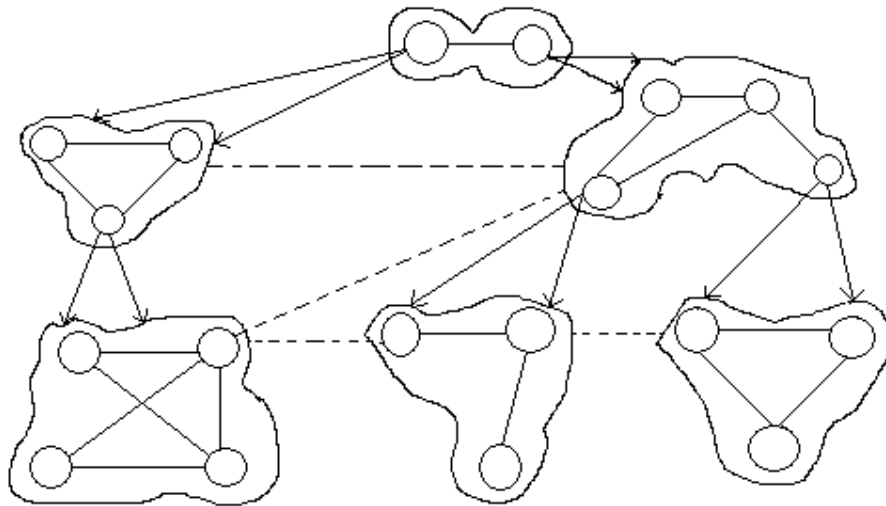
CHƯƠNG 3 CÔNG NGHỆ PHẦN MỀM HƯỚNG TÁC TỬ

3.1 TIẾP CẬN HƯỚNG TÁC TỬ CHO CÔNG NGHỆ PHẦN MỀM

Phần mềm công nghiệp về bản chất là phức tạp: nó được mô tả bởi một số lượng lớn các thành phần trong đó có nhiều tương tác. Hơn nữa, sự phức tạp này lại không ngẫu nhiên: nó là thuộc tính bẩm sinh của các kiểu công việc mà phần mềm được sử dụng. Bởi thế vai trò của công nghệ phần mềm là cung cấp các cấu trúc về kỹ thuật để quản lý sự phức tạp này dễ dàng hơn. Sự phức tạp này có những đặc điểm quan trọng: Sự phức tạp thường có một hình dáng phân cấp. Hệ thống được tạo thành bởi các hệ thống con lại có thể là một cây phân cấp. Tính chất của các mối quan hệ thuộc tổ chức thay đổi giữa các hệ thống con, mặc dù một số kiểu (chẳng hạn như client-server, peer, team) có thể được xác định. Các mối quan hệ thuộc tổ chức là không tĩnh: chúng có thể thay đổi thường xuyên theo thời gian. Sự lựa chọn các thành phần nguyên thủy là tùy ý quan hệ và được xác định bởi các mục đích và mục tiêu của người quan sát là chủ yếu. Các hệ thống phân cấp tiến hoá nhanh hơn so với các hệ thống không phân cấp có cùng kích thước. Nói cách khác, các hệ thống phức tạp sẽ tiến hoá từ các hệ thống đơn giản nhanh hơn khi có các mẫu trung gian ổn định. Có thể phân biệt các tương tác *giữa* các hệ thống con và các tương tác *trong* các hệ thống con. Điều này cho thấy các hệ thống phức tạp là có thể phân rã. Bởi vậy, các hệ thống có thể được xử lý như là chúng độc lập với nhau, nhưng cũng không hoàn toàn như vậy nếu như có các tương tác ở giữa chúng. Hơn nữa, mặc dù nhiều trong số các tương tác đó có thể dự đoán được ở giai đoạn thiết kế nhưng một số khác thì không.

Có thể định nghĩa một cách nhìn đúng đắn về một hệ thống phức tạp bằng cách đưa ra cách nhìn xuyên suốt bên trong như (hình vẽ). Tính chất phân cấp của

hệ thống được thể hiện qua các kết nối kiểu “composed of”, các thành phần ở trong một hệ thống con được kết nối với nhau thông qua các kết nối “frequent interaction” và các tương tác giữa các thành phần được thể hiện qua các liên kết “infrequent interaction”. Khái niệm các thành phần nguyên thủy thay đổi có thể được coi như cách các thành phần nguyên tố ở mức 1 được mở rộng ra các hệ thống con còn lại ở các mức sau.



Hình 3.1 Hệ thống phần mềm phức tạp

Từ các quan sát này, các kỹ sư phần mềm đã đưa ra một số công cụ mạnh để quản lý sự phức tạp này. Các kỹ thuật chủ yếu bao gồm:

- *Decomposition*: Kỹ thuật cơ bản nhất cho việc xử lý một bài toán lớn để phân chia nó thành nhỏ hơn, các phân đoạn dễ quản lý hơn và sau đó có thể được xử lý trong một tổ chức quan hệ (cho thấy ở các hệ thống con có thể phân rã được trong hình vẽ). Sự phân rã trợ giúp giải quyết sự phức tạp bởi vì nó giới hạn phạm vi của người thiết kế: ở một thời điểm nào đó, chỉ một phần của bài toán cần được xemt.

- *Abstraction*: Các định nghĩa một mô hình của hệ thống được đơn giản hoá bằng cách chỉ nhấn mạnh một số chi tiết và thuộc tính. Một lần nữa, kỹ thuật hoạt động vì nó hạn chế phạm vi quan tâm của nhà thiết kế ở một thời điểm nào đó. Sự

chú ý có thể được tập trung trên các hướng nổi bật nhất của vấn đề, ở mức các chi tiết ít liên quan nhất.

- *Organisation*: Cách nhận dạng và kiểm soát các mối quan hệ bên trong giữa các thành phần khác nhau của vấn đề đang được giải quyết. Khả năng để chỉ rõ và đưa ra các mối quan hệ thuộc tổ chức trợ giúp người thiết kế kiểm soát được sự phức tạp theo hai cách. Thứ nhất, cho phép một số lượng các thành phần cơ bản được nhóm lại với nhau và được xử lý như ở một đơn vị phân tích mức cao hơn. Ví dụ, các thành phần đơn lẻ của một hệ thống con có thể được xử lý như là một đơn vị đơn lẻ bởi hệ thống cha. Thứ hai, bằng cách cung cấp một phương tiện mô tả các mối quan hệ mức cao giữa các đơn vị khác nhau. Chẳng hạn như một số các thành phần có thể cần để hoạt động cùng nhau để cung cấp một chức năng cụ thể nào đó. Tính chất đúng đắn và cách mà các công cụ này được sử dụng thay đổi nhiều giữa các mô hình phần mềm. Vì thế, khi mô tả đặc điểm của một mô hình mới, chẳng hạn như phần mềm hướng tác tử, ở mỗi một vấn đề đưa ra thì các quan điểm cần được xác định rõ ràng. Hơn nữa, khi đánh giá khả năng của một mô hình, các tham số cần được đề xuất tại sao cách lựa chọn để xử lý các vấn đề trợ giúp các kỹ sư phần mềm xây dựng các hệ thống hiệu quả hơn.

3.2 PHẦN MỀM HƯỚNG TÁC TỬ LÀ GÌ?

Hiện nay, có rất nhiều cuộc tranh luận đang xảy ra xung quanh định nghĩa chính xác về cái gì tạo thành một tác tử, nhưng vẫn chưa có một sự nhất trí hoàn toàn nào. Tuy nhiên, các nhà nghiên cứu tìm ra sự mô tả đặc điểm hữu ích sau: *“Một tác tử là một hệ thống máy tính được đóng gói và đặt trong một vài môi trường và có khả năng hành động mềm dẻo, tự trị trong môi trường đó để đạt được các mục tiêu thiết kế của nó”*.

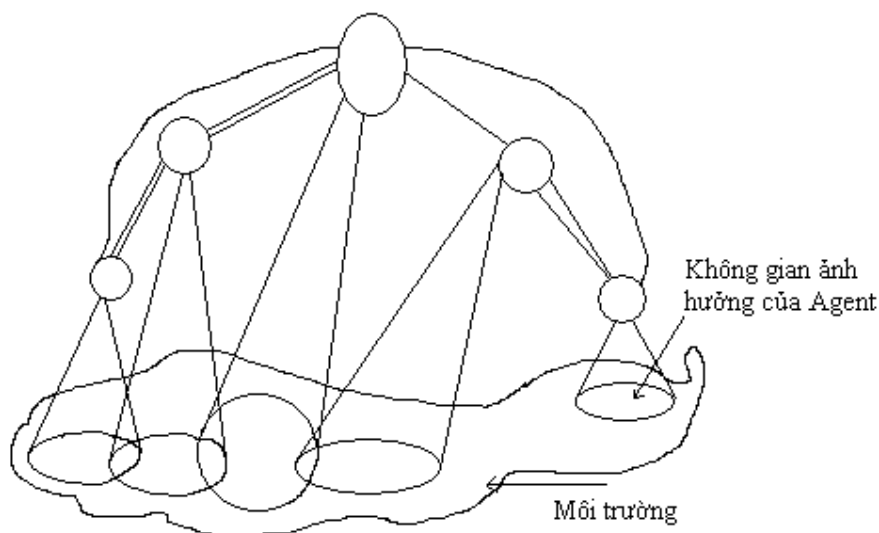
Có một số vấn đề định nghĩa này cần giải thích rõ hơn. Các tác tử là: các thực thể có thể nhận dạng rõ ràng của bài toán đang được giải quyết với các giao diện và ranh giới rõ ràng; được đặt trong một môi trường cụ thể- chúng nhận các đầu vào từ trạng thái của môi trường thông qua các bộ cảm nhận của chúng và tác động lên môi trường thông qua cơ quan cảm ứng của chúng; được thiết kế để thực

hiện một vai trò cụ thể- chúng có các mục tiêu cụ thể cần đạt đến, có thể rõ ràng hay không rõ ràng trong các tác tử; tự trị-chúng kiểm soát qua trạng thái trong và các hành vi của chúng; Có khả năng thể hiện hành vi giải quyết vấn đề mềm dẻo (phụ thuộc ngữ cảnh)- chúng cần *reactive* (có thể phản ứng ngay lập tức với những thay đổi của môi trường để thoả mãn các mục tiêu thiết kế của chúng) và *proactive* (có khả năng cập nhật các mục đích mới và nắm thể chủ động để thoả mãn mục tiêu thiết kế của chúng).

Khi nhìn nhận thể giới theo kiểu hướng tác tử, sẽ sớm rõ ràng rằng một tác tử đơn lẻ là không đủ. Hầu hết các bài toán đòi hỏi hoặc là liên quan đến đa tác tử: để thực hiện tính chất phân quyền của bài toán, đa quỹ tích của điều khiển, đa viễn cảnh hay các quan tâm cạnh tranh. Hơn nữa, các tác tử sẽ cần để tương tác với các tác tử khác, để đạt được các mục tiêu riêng của chúng hoặc để điều khiển các phụ thuộc sinh ra từ các trạng thái trong môi trường chung. Các tương tác này xuyên suốt từ các thao tác ngữ nghĩa đơn giản (khả năng thay đổi các giao tiếp toàn diện) qua các tương tác kiểu client-server truyền thống (khả năng yêu cầu một hành động cụ thể được thực hiện), tới các tương tác xã hội (khả năng phối hợp, điều phối và thương thuyết về cách thức của hành động). Tuy nhiên, bất cứ tính chất nào của quá trình xã hội cũng có hai điểm tương tác tác tử khác biệt định tính mà từ đó xuất hiện ở các mô hình công nghệ phần mềm khác. Thứ nhất, các tương tác hướng tác tử thường xảy ra ở ngôn ngữ giao tiếp tác tử bậc cao (điển hình dựa vào lý thuyết hành vi ngôn ngữ). Do đó, các tương tác thường là được điều khiển ở mức tri thức: trong các giới hạn mà các mục đích nên theo. Thứ hai, vì tác tử là giải pháp mềm dẻo cho bài toán, việc thao tác trên một môi trường mà qua đó chúng chỉ có quan sát và điều khiển cục bộ, các tương tác cần được xử lý trong một kiểu mềm dẻo tương tự, bởi vậy các tác tử cần dụng cụ tính toán để đưa ra các quyết định phụ thuộc vào ngữ cảnh về tính chất và phạm vi của các tương tác của chúng và để bắt đầu (hoặc đáp ứng) các tương tác mà không cần thiết được dự đoán trước ở giai đoạn thiết kế.

Trong hầu hết các trường hợp, các tác tử hành động để đạt đến mục tiêu đại diện cho các thành phần riêng lẻ hay các hệ thống. Vì vậy, khi các tác tử tương tác,

thường có một số ngữ cảnh cơ bản thuộc tổ chức. Ngữ cảnh này trợ giúp việc định nghĩa tính chất của mối quan hệ giữa các tác tử. Ví dụ, chúng có thể hoạt động ngang hàng trong một nhóm, một tác tử có thể là ông chủ của các tác tử khác, hoặc chúng có thể được bao hàm trong một loạt các mối quan hệ chủ-tớ. Để giành được các liên kết đó, các hệ thống tác tử thường có cấu trúc rõ ràng cho việc mô hình các mối quan hệ tổ chức (chẳng hạn như peer,boss,..) và các cấu trúc tổ chức(chẳng hạn như là teams,groups,...). Nên lưu ý rằng trong nhiều trường hợp, các mối quan hệ này có thể thay đổi trong khi hệ thống hoạt động. Tương tác xã hội có nghĩa là sự tồn tại các mối quan hệ tiến hoá (chẳng hạn một tác tử quyết định một giao dịch mới) và các mối liên hệ mới được tạo ra (chẳng hạn như một số tác tử có thể hình thành một nhóm để đưa ra một dịch vụ cụ thể mà không một cá nhân nào có thể đề nghị). Phạm vi thời gian của các mối quan hệ đó có thể được thay đổi nhiều: từ chỗ chỉ đủ dài để đưa ra một dịch vụ cụ thể một lần tới một liên kết lâu dài. Để đương đầu với sự thay đổi về biến động này, các nhà nghiên cứu tác tử đã có những nỗ lực đáng kể: phát minh ra các giao thức mà cho phép các nhóm tổ chức được hình thành và giải tán, chỉ rõ các cơ chế để bảo đảm các nhóm hoạt động cùng nhau trong mô hình kết dính, và phát triển các cấu trúc để mô tả đặc điểm các hành vi vĩ mô của các tập hợp.



Hình 3.2 Quan điểm kinh điển hệ thống đa-tác tử

Bằng cách vẽ các vấn đề trên cùng nhau, ta có thể thấy việc chấp nhận một hướng tiếp cận hướng tác tử với công nghệ phần mềm có nghĩa là việc phân rã bài toán thành nhiều thành phần tương tác và tự trị (tác tử) mà có các mục tiêu cụ thể đạt tới. Các mô hình trừu tượng chủ yếu định nghĩa “agent-oriented mindset” là các tác tử, các tương tác và các tổ chức. Cuối cùng, các cấu trúc và cơ chế rõ ràng thường là có sẵn cho việc mô tả và điều khiển sự phức tạp và thay đổi một mạng các mối quan hệ tổ chức mà tồn tại giữa các tác tử.

3.3 TIẾP CẬN HƯỚNG TÁC TỬ CHO CÁC HỆ THỐNG PHẦN MỀM

Từ các mô tả về các hệ thống phức tạp và phần mềm hướng tác tử ở trên, chúng ta xem xét tại sao các kỹ thuật hướng tác tử lại phù hợp với việc phát triển các hệ thống phần mềm này. Vấn đề có 3 phần:

- Chỉ ra rằng các phân rã hướng tác tử là một cách hiệu quả để phân chia không gian bài toán của một hệ thống phức tạp.
- Chỉ ra rằng các sự trừu tượng chủ yếu của mindset hướng tác tử là một cách tự nhiên để mô hình hoá các hệ thống phức tạp.
- Chỉ ra rằng triết lý hướng tác tử cho việc nhận dạng và điều khiển các mối quan hệ là thích hợp cho việc xử lý với các phụ thuộc và tương tác tồn tại trong một hệ thống phức tạp.

Để làm cho công nghệ phần mềm hướng tác tử thuyết phục hơn, bước cuối cùng là chúng ta chứng tỏ rằng các kỹ thuật hướng tác tử thể hiện một sự tiến bộ thật sự qua tình trạng hiện thời của kỹ thuật. Với mục đích này, tiếp cận hướng tác tử sẽ được so sánh với các kỹ thuật hướng lề (leading-edge) từ công nghệ phần mềm xu hướng chính. Đặc biệt, nó bao hàm các sự so sánh với phân tích và thiết kế hướng đối tượng (hệ thống được xây dựng bằng các đối tượng tương tác mà đóng gói cả dữ liệu và thủ tục thao tác trên dữ liệu) và với component-ware (hệ thống được xây dựng bởi việc tập hợp các thành phần tồn tại sẵn vào một số cấu trúc tổng thể).

3.3.1 Các phân rã hướng tác tử

Các hệ thống phức tạp bao gồm các hệ thống con có liên quan được tổ chức lại với nhau trong một mô hình phân cấp (hình 3.1). Ở một mức độ nào đó, các hệ thống con hoạt động cùng nhau để đạt đến chức năng của hệ thống cha của chúng. Hơn nữa, trong một hệ thống con, các thành phần hợp thành hoạt động cùng nhau để đưa ra chức năng tổng thể. Bởi vậy, mô hình cơ bản tương tự của các thành phần tương tác hoạt động cùng nhau để đạt tới các mục tiêu cụ thể xuất hiện trong hệ thống.

Từ trạng thái được đưa ra trên đây, việc modul hoá các thành phần là hoàn toàn tự nhiên trong giới hạn các mục tiêu mà chúng cần đạt đến. Nói cách khác, mỗi thành phần có thể được nghĩ về như là việc đạt tới một hay nhiều mục tiêu. Một quan sát quan trọng thứ hai là các hướng công nghệ phần mềm hiện tại đang tăng mức độ của cục bộ hoá và gói gọn trong các phân rã bài toán. Việc ứng dụng triết lý này vào các phân rã các mục tiêu đạt tới có nghĩa là các thành phần đơn lẻ nên có một thành phần điều khiển riêng của chúng (tức là các thành phần nên chủ động) và chúng nên gói gọn thông tin và cần khả năng giải quyết vấn đề để đạt được các mục tiêu này. Vì các thành phần thường là phải thao tác trong một môi trường mà trong đó chúng chỉ có một phần thông tin, chúng phải có khả năng quyết định ở thời gian chạy hành động nào nên thực hiện để đuổi theo các mục tiêu của chúng. Tóm lại, các thành phần cần tự trị qua các lựa chọn hành động của chúng.

Với các thành phần chủ động và tự trị để thực hiện các mục tiêu cá nhân và tập thể, chúng cần phải tương tác với nhau (lưu ý rằng các hệ thống phức tạp là có thể được phân rã). Tuy nhiên sự phức tạp của hệ thống là vốn có, nghĩa là nó không thể biết về tất cả các liên hệ trong tương lai: tương tác sẽ xảy ra ở một thời điểm không thể đoán, với các lý do không thể đoán được, giữa các thành phần không thể dự đoán. Vì lý do này, việc cố gắng để dự đoán hoặc phân tích tất cả khả năng trong giai đoạn thiết kế là vô ích. Thực tế hơn là cung cấp cho các thành phần một khả năng đưa ra các quyết định về tính chất và phạm vi của các tương tác ở thời gian chạy. Từ đó dẫn đến các thành phần cần một khả năng để khởi sướng (và đáp ứng) các tương tác một cách mềm dẻo.

Cách giải quyết làm theo các quyết định thi hành về các tương tác thành phần làm cho việc thực hiện các hệ thống phức tạp một cách dễ dàng trong hai cách. Thứ nhất, các vấn đề liên quan đến sự kết nối của các thành phần giảm một cách đáng kể (bằng việc xử lý chúng theo cách mềm dẻo và khai báo). Cách thành phần được thiết đặc biệt để giải quyết với các yêu cầu không thể đoán trước và có thể đưa ra các yêu cầu hỗ trợ nếu chúng gặp khó khăn. Hơn nữa, vì các tương tác này được đưa ra qua một mức ngôn ngữ giao tiếp bậc cao, sự kết nối trở thành một vấn đề mức tri thức. Điều này lần lượt loại bỏ những quan tâm ở mức cú pháp từ các kiểu của lỗi được gây ra bởi các tương tác không mong đợi. Thứ hai, vấn đề kiểm soát điều khiển các mối quan hệ giữa các thành phần phần mềm bị giảm một cách đáng kể. Tất cả tác tử liên tục hoạt động và một số điều phối hoặc sự đồng bộ hoá được yêu cầu được điều khiển qua tương tác trong tác tử. Vì vậy trật tự của các mục đích mức đỉnh của hệ thống không phải được quy định một cách cứng nhắc ở giai đoạn thiết kế. Đúng hơn, nó thể xử lý theo các tác động của ngữ cảnh ở thời gian chạy.

Từ lý do này, một cách tự nhiên để modul hóa một hệ thống phức tạp trong một giới hạn của nhiều thành phần tương tác, tự trị có các mục tiêu cụ thể đạt tới là đã rõ ràng. Tóm lại, các phân rã hướng tác tử làm cho việc phát triển các hệ thống phức tạp dễ dàng hơn.

3.3.2 Các trừu tượng hoá hướng tác tử cho các hệ thống phần mềm phức tạp

Tất cả các nỗ lực thiết kế chủ yếu là để tìm ra các mô hình mô tả bài toán. Nhìn chung, sẽ có nhiều ứng cử viên và nhiệm vụ khó khăn là chọn ra ứng cử viên thích hợp nhất. Trở lại với trường hợp cụ thể của việc thiết kế phần mềm, các trừu tượng hoá mạnh nhất giảm thiểu được các thiếu sót về ngữ nghĩa giữa các đơn vị của phân tích mà được sử dụng bởi trực giác để khái niệm hoá một vấn đề và các cấu trúc thể hiện trong mô hình giải quyết. Trong trường hợp này, vấn đề được mô tả bao gồm các hệ thống con, các tương tác và các mối quan hệ thuộc tổ chức. Lần lượt như sau:

- Trường hợp quan sát các thành phần hệ thống con như là tác tử được tạo ra ở trên.
- Sự ảnh hưởng lẫn nhau giữa các hệ thống con và giữa các thành phần tạo thành các hệ thống con được coi là cơ bản nhất trong giới hạn của các tương tác xã hội mức cao: “Ở một mức độ trừu tượng cho trước, ta tìm thấy tập hợp các đối tượng mà cộng tác với nhau để đạt được các quan điểm mức cao hơn”. Quan điểm này phù hợp với cách quản lý mức tri thức (hoặc thậm chí mức xã hội) của tương tác được đưa ra bởi tiếp cận hướng tác tử. Các hệ thống tác tử được mô tả cố định trong giới hạn của “sự phối hợp để đạt được các mục đích chung”, “sự điều phối các hành động của chúng” hoặc “sự thương lượng để giải quyết xung đột”. Vì vậy, mindset hướng tác tử là hoàn toàn phù hợp cho việc kiểm soát các kiểu tương tác xảy ra trong các hệ thống phức tạp.
- Các hệ thống phức tạp bao gồm sự thay đổi các mạng quan hệ giữa các thành phần khác nhau. Chúng cũng yêu cầu các tập hợp các thành phần được xử lý như là một đơn vị khái niệm đơn lẻ khi được quan sát từ một mức độ trừu tượng khác. Một lần nữa, quan điểm này phù hợp với sự trừu tượng hoá được cung cấp bởi mindset hướng tác tử. Vì vậy, các phương tiện thường là được cung cấp cho việc biểu diễn rõ ràng các mối quan hệ thuộc tổ chức. Các giao thức tương tác được phát triển để hình thành các nhóm mới và huỷ những nhóm không cần thiết. Cuối cùng, các cấu trúc là sẵn có cho việc mô hình hoá các tập hợp. Điểm sau đó là đặc biệt có ích trong mối quan hệ với việc biểu diễn các hệ thống con vì chúng chỉ là một nhóm các thành phần hoạt động cùng nhau để đạt đến một mục đích chung.

3.3.3 Sự thay đổi các cấu trúc trong tổ chức tạo quản lý mềm dẻo

Các hệ thống phức tạp có các mối quan hệ tổ chức đa dạng phong phú, xuyên suốt từ các thành phần ngang hàng tới việc điều khiển các cây phân cấp, từ ngắn hạn đến liên tục. Các mối quan hệ này là quan trọng vì hai lý do. Thứ nhất, chúng cho phép một số các thành phần riêng lẻ được gom nhóm lại với nhau và xử lý như

là một thực thể khái niệm đơn lẻ. Thứ hai, chúng cho phép mô tả đặc điểm các môi liên kết mức cao giữa các thực thể khác nhau. Sự ảnh hưởng và tác động đã biết về các môi quan hệ và các cấu trúc tổ chức trên hành vi của hệ thống, sự quan trọng của việc cung cấp hỗ trợ rõ ràng cho việc chỉ rõ và điều khiển mềm dẻo là bản thân một bằng chứng. Hơn nữa, các môi quan hệ thì liên tục thay đổi, khả năng để thích nghi một cách linh động để nắm ưu thế các hoàn cảnh cũng là cần thiết.

Như đã đưa ra ở trên, các tổ chức là các thực thể ở lớp đầu tiên trong các hệ thống tác tử. Vì vậy, các cấu trúc rõ ràng và các cơ chế mềm dẻo là trung tâm cho mô hình tác tử. Khả năng biểu diễn này khi được liên kết với các cơ chế tính toán hỗ trợ, cho phép các hệ thống hướng tác tử phân rã thành hai khía cạnh của tính chất các hệ thống phức tạp. Thứ nhất, khái niệm của thành phần nguyên thủy có thể thay đổi theo nhu cầu của người quan sát. Do đó ở mức 1, các hệ thống còn lại có thể được coi như là một singleton (một thực thể duy nhất), các nhóm hay tập hợp thay thế của các tác tử có thể được coi như là các thành phần nguyên thủy và cứ như vậy cho đến khi hệ thống cuối cùng được phân rã hết. Thứ hai, các cấu trúc này cung cấp một sự đa dạng các forms trung gian ổn định cần thiết cho sự phát triển nhanh các hệ thống phức tạp. Tính sẵn có của chúng có nghĩa là các tác tử đơn lẻ hoặc nhóm tổ chức có thể được phát triển trong sự cô lập về quan hệ và sau đó được thêm vào hệ thống theo cách thức tăng trưởng. Điều này lần lượt đảm bảo rằng có một sự làm mịn chức năng.

3.4 VÒNG ĐỜI PHẦN MỀM HƯỚNG TÁC TỬ

Chúng ta đã hiểu tại sao các kỹ thuật hướng tác tử thể hiện một cách tiếp cận đầy hứa hẹn cho việc xây dựng các hệ thống phức tạp, chúng ta có thể đi vào chi tiết của công nghệ phần mềm hướng tác tử, thảo luận về cách thực hiện các đặc tả này và khảo sát tỉ mỉ cách xác nhận các hệ thống đã được thực hiện trong thực tế là thoả mãn các đặc tả của chúng.

3.4.1 Đặc tả (Specification)

Trong phần này chúng ta xem xét bài toán đặc tả một hệ thống tác tử. Các yêu cầu cho một khung đặc tả tác tử. Các kiểu thuộc tính có thể được thể hiện. Theo

cách nhìn các tác tử được thảo luận ở trên, cách tiếp cận nổi bật để đặc tả các tác tử bao gồm cả việc xử lý chúng như là các hệ thống có mục đích có thể được hiểu bằng cách quy cho chúng các trạng thái tri thức chẳng hạn như lòng tin, mong muốn và mục đích. Dựa vào ý tưởng này, một số cách tiếp cận cho việc đặc tả một cách hình thức các tác tử đã được phát triển, có khả năng thể hiện các hướng sau đây của một hệ thống tác tử:

- *Beliefs*: thông tin mà các tác tử có về môi trường xung quanh nó, có thể không đầy đủ hoặc không chính xác.
- *Goals*: là các mục đích mà các tác tử cố gắng đạt đến;
- *Action*: các hành động mà các tác tử thực hiện và các ảnh hưởng của các hành động đó
- *Ongoing interaction*: cách mà tác tử tương tác với cá tác tử khác trong môi trường của chúng qua thời gian.

Ta gọi lý thuyết giải thích các hướng tương tác kiểu tác tử để đạt đến ánh xạ từ đầu vào và đầu ra là một lý thuyết tác tử. Lý thuyết về tác tử thành công nhất là ứng dụng của một *temporal modal logic* (các giới hạn không gian ngăn chặn sự thảo luận kỹ thuật chi tiết trên các logic-see, chẳng hạn như cho các tham khảo rộng rãi). Hai trong số các khung logic nổi tiếng nhất là lý thuyết Cohen-Levesque về mục đích và mô hình Rao-Georgeff belief-desire-intention. Mô hình Cohen-Levesque về lúc đầu chỉ có hai quan điểm: belief và goals. Các quan điểm khác (cụ thể là khái niệm intention) được xây dựng từ đó. Ngược lại, Rao-Georgeff lại coi intention là cơ bản ban đầu, thêm vào đó là belief và goals. Vấn đề kỹ thuật chính của các nhà lý thuyết tác tử là việc phát triển một mô hình chính thức có thể đưa ra một lời giải thích phù hợp cho các mối quan hệ bên trong giữa các quan điểm khác nhau tạo nên một tác tử. Các nỗ lực tương đối ít quan trọng hơn được đưa ra nhằm chỉ rõ các hệ thống tác tử thực sự sử dụng logics-see.

Một khung đặc tả tác tử modal temporal điển hình bao gồm:

- Các kết nối logic hình thức chuẩn cho việc thể hiện các beliefs của tác tử;

- Các kết nối logic thời gian cho việc thể hiện về xử lý động của hệ thống-các hành vi được phát triển liên tục của nó;
- Các kết nối logic hình thức chuẩn cho việc thể hiện các mục đích (chẳng hạn như desires, intention, obligations);
- Một số các bộ phận quan trọng cho việc thể hiện các hành động mà các tác tử thực hiện.

Từ các yêu cầu đã cho này, có nhiều chiều hướng mà một khung đặc tả tác tử có thể thay đổi, một số hướng được tổng hợp trong bảng sau:

<i>Các khía cạnh thông tin:</i> <i>Tri thức</i> <i>Lòng tin</i> <i>Tập hợp các quan điểm thông tin</i>
<i>Các khía cạnh thời gian:</i> <i>Tuyến tính đối với phân nhánh</i> <i>Dày đặc đối với rời rạc</i> <i>Tham khảo trực tiếp đối với cá thao tác căng thẳng</i> <i>Dựa điểm đối với dựa vào khoảng</i>
<i>Các khía cạnh mục đích:</i> <i>Mong muốn</i> <i>Mục đích</i> <i>Nghĩa vụ</i> <i>Lựa chọn</i> <i>Tập các quan điểm thay thế</i>
<i>Các hành động</i> <i>Biểu diễn trực tiếp</i> <i>Biểu diễn ẩn</i>

3.4.2 Thực hiện (Implementation)

Một khi đã có một đặc tả, ta phải thực hiện một hệ thống đúng đắn với đặc tả này. Vấn đề tiếp theo là sự chuyển đổi từ một đặc tả trừu tượng sang một hệ thống tính toán cụ thể. Ở đây ta xem xét hai khả năng để thực hiện sự chuyển đổi này:

- Làm mịn thủ công đặc tả thành một form có thể thực hiện được theo các nguyên lý nào đó, nhưng không phải là một quá trình làm mịn chính thức (như trong hầu hết sự phát triển phần mềm hiện nay).
- Bằng cách nào đó thực hiện trực tiếp hoặc hoạt hoá đặc tả trừu tượng;
- Hoặc bằng cách nào đó chuyển đổi hoặc biên dịch đặc tả thành mẫu tính toán cụ thể sử dụng một kỹ thuật chuyển đổi tự động.

3.4.2.1 Làm mịn (Refinement)

Hầu hết các nhà phát triển phần mềm sử dụng các kỹ thuật cấu trúc không chính thức để chuyển các đặc tả thành các thực hiện cụ thể. Hầu hết các kỹ thuật được ứng dụng rộng rãi dựa vào ý tưởng làm mịn top-down. Trong cách tiếp cận này, một đặc tả hệ thống trừu tượng được làm mịn thành các đặc tả hệ thống con nhỏ hơn và ít trừu tượng hơn mà cùng nhau thoả mãn đặc tả ban đầu. Nếu các hệ thống con này vẫn còn quá trừu tượng cho việc thực hiện trực tiếp thì chúng vẫn tiếp tục được làm mịn. Từ đầu đến cuối, ta bị bắt buộc phải chứng minh rằng mỗi bước thể hiện một sự làm mịn đúng đắn của đặc tả trừu tượng hơn được đặt trước nó. Sự chứng minh này có thể đưa ra form chứng minh hình thức nếu như đặc tả của ta được thể hiện trong đó gọi là Z hoặc là VDM. Sự chứng minh bằng tham số không chính thức là thường dùng hơn.

Với các hệ thống chức năng, quá trình làm mịn được hiểu rõ và tương đối dễ hiểu. Tồn tại các công thức làm mịn cho phép người phát triển hệ thống đưa ra các đặc tả trước và sau điều kiện mà mô tả đặc điểm các cấu trúc thao tác và chương trình được yêu cầu để thực hiện.

Với các hệ thống phản xạ, việc làm mịn là không quá dễ dàng, vì các hệ thống phản xạ phải được chỉ rõ trong giới hạn của hành vi đang phát triển của chúng. Ngược lại, với các cơ chế hình thức trước và sau điều kiện thì không dễ để quyết định các cấu trúc chương trình nào được yêu cầu thực hiện các đặc tả này. Bài

toán làm mịn cho các hệ thống hướng tác tử có các đặc tả có thể được đánh giá là thậm chí trừu tượng hơn với các hệ thống phân xạ là khó khăn. Từ đó, các nhà nghiên cứu chỉ mới bắt đầu khảo sát cách làm mịn các hệ thống dựa tác tử.

Một phương pháp luận cho các tác tử BDI

Mô hình BDI như đã nói ở trên là một trong những khung thành công nhất cho tác tử. Kiny và các đồng nghiệp đã giả sử một phương pháp thiết kế bốn giai đoạn cho các hệ thống các tác tử BDI. Phương pháp này liên kết gắn gũi với một sự thực hiện cụ thể của mô hình BDI: kiến trúc PRS. Có thể tổng hợp phương pháp này theo các bước sau:

- Nhận ra các vai trò liên quan trong lĩnh vực ứng dụng và từ đó phát triển một cây phân cấp lớp tác tử. Một vai trò ví dụ có thể là theo dõi thời tiết, trong đó tác tử I được yêu cầu để làm cho tác tử J biết được điều kiện thời tiết trong từng giờ.
- Xác định các trách nhiệm liên quan với mỗi vai trò, các dịch vụ được yêu cầu và được cung cấp bởi vai trò đó và sau đó quyết định mục đích liên quan với mỗi dịch vụ. Từ đó các mục đích có thể để tìm ra tình trạng thời tiết hiện tại và làm cho tác tử j biết được các thông tin này.
- Với mỗi mục đích, quyết định các kế hoạch có thể được sử dụng để đạt được nó và dưới các điều kiện ngữ cảnh mà mỗi kế hoạch thoả mãn. Từ đó, một kế hoạch cho mục đích làm cho tác tử j hiểu được các điều kiện thời tiết có thể bao gồm việc gửi một thông điệp tới j .
- Xác định cấu trúc niềm tin của hệ thống-các yêu cầu thông tin cho mỗi kế hoạch và mục đích. Từ đó ta có thể đưa ra một hàm dự đoán $windspeed(x)$ để biểu diễn tốc độ gió hiện tại là x . Một kế hoạch cho việc xác định các điều kiện thời tiết hiện tại có thể cần thiết để biểu diễn thông tin này.

Lưu ý rằng quá trình phân tích sẽ được lặp đi lặp lại như trong các phương pháp truyền thống hơn. Đầu ra sẽ là một mô hình gần tương đương với kiến trúc của tác tử PRS. Từ đó, việc chuyển từ bản thiết kế cuối cùng tới việc thực hiện sử dụng PRS là tương đối đơn giản.

Kinny và các cộng sự minh hoạ phương pháp của họ bằng việc ứng dụng nó vào việc thực hiện hệ thống quản lý giao thông trên không OASIS.

3.4.2.2 Việc thực hiện trực tiếp các đặc tả tác tử

Giả sử rằng chúng ta có một đặc tả hệ thống ϕ được biểu diễn trong một số ngôn ngữ logic L. Một cách để đạt được một hệ thống cụ thể từ ϕ là coi nó như là một đặc tả có thể thực hiện được, và thông dịch đặc tả một cách trực tiếp để tạo ra hành vi của tác tử. Việc thông dịch một đặc tả tác tử có thể được xem như là một kiểu chứng minh suy diễn của sự thoả mãn, nhờ đó ta thấy được đặc tả ϕ là thoả mãn việc xây dựng một mô hình (về ý nghĩa logic) cho nó. Nếu các mô hình với ngôn ngữ đặc tả L có thể được đưa ra một sự thông dịch tính toán, sau đó việc xây dựng mô hình có thể được coi như là việc thực hiện đặc tả. Để làm rõ hơn vấn đề này, ta xem xét ngôn ngữ lập trình Concurrent Metate M. Trong ngôn ngữ này, các tác tử được lập trình bằng cách trao cho chúng một đặc tả logic thời gian của hành vi, chúng được lưu ý là nên thể hiện; đặc tả này được thực hiện trực tiếp để tạo ra mỗi hành vi của tác tử. Vì vậy, các mô hình cho logic thời gian trong đó đặc tả tác tử MetateM hiện tại là quá trình của việc xây dựng một chuỗi các trạng thái. Do đó chuỗi các trạng thái có thể được coi như là các lịch sử được xác định bằng các chương trình khi chúng thực hiện, logic thời gian dựa vào Concurrent MetateM có một sự thông dịch tính toán.

Lưu ý rằng việc thực hiện các đặc tả tác tử Concurrent MetateM có thể là cơ bản vì các mô hình theo logic thời gian có nền tảng tương đối đơn giản, với một sự thông dịch tính toán rõ ràng và thuộc về trực giác. Tuy nhiên các ngôn ngữ đặc tả tác tử nhìn chung là được dựa trên các logic phức tạp hơn (chẳng hạn như cơ chế hình thức BDI của Rao và Georgeff). Đặc biệt chúng thường dựa trên một khung ngữ nghĩa được coi như là *possible worlds*.

3.4.2.3 Việc biên dịch các đặc tả tác tử

Một sự thay đổi cho việc thực hiện trực tiếp là sự biên dịch (*compilation*). Theo ý đồ này, ta biết được đặc tả trừu tượng và chuyển đổi chúng thành một mô

hình tính toán cụ thể qua một số quá trình tổng hợp tự động. Có thể nhận thấy các ưu điểm của sự biên dịch qua việc thực hiện trực tiếp trong hiệu xuất thi hành. Việc thực hiện trực tiếp của một đặc tả tác tử, chẳng hạn như trong Concurrent MetateM ở trên điển hình bao gồm sự thực hiện một biểu diễn tượng trưng của đặc tả ở thời gian chạy. Sự thực hiện này tương đương với việc suy diễn của một số mẫu đáng giá tính toán. Các hướng tiếp cận biên dịch giúp giảm bớt các đặc tả trừu tượng thành một mô hình tính toán đơn giản hơn mà không yêu cầu sự biểu diễn tượng trưng. Việc suy diễn bởi vậy được thực hiện off-line ở thời điểm biên dịch; việc thực hiện của hệ thống được biên dịch sau đó có thể được thực hiện với rất ít hoặc không có suy diễn tượng trưng ở thời điểm thi hành.

Stituated Automata: Các máy tự động trạng thái

Các hướng tiếp cận biên dịch thường dựa vào mối quan hệ thân thiết giữa các mô hình logic thời gian/hình thức, và các máy trạng thái giống các máy tự động. Ví dụ, Pnueli và Rosner tổng hợp các hệ thống phản xạ từ việc phân chia các đặc tả logic thời gian. Các kỹ thuật tương tự cũng đã được sử dụng để phát triển các nhân hệ thống trùng hợp từ các đặc tả logic thời gian. Có thể ví dụ điển hình nhất cho hướng tiếp cận này của việc phát triển tác tử là mô hình các máy tự động trạng thái (stituated automata). Họ sử dụng một logic thuộc tri thức (tức là một logic của nhận thức) để chỉ rõ thành phần nhận thức của các hệ thống tác tử thông minh. Sau đó họ sử dụng một kỹ thuật dựa vào chứng minh suy diễn để tổng hợp trực tiếp các máy tự động từ các đặc tả này.

Trong hướng tiếp cận các máy tự động trạng thái, một tác tử có 2 thành phần chính:

- Một bộ phận tri giác (*perception*) có trách nhiệm quan sát môi trường và cập nhật trạng thái trong của tác tử.
- Một bộ phận hành động (*action*) có trách nhiệm quyết định các hành động để thực hiện, dựa trên trạng thái bên trong của tác tử.

Rosenschein và Kaelbling đã phát triển hai chương trình tương ứng để hỗ trợ sự phát triển của các thành phần tri giác và hành động của tác tử.

Hướng tiếp cận chung của sự tổng hợp tự động, mặc dù hấp dẫn về lý thuyết nhưng lại bị giới hạn trong một số lượng của các mối quan hệ quan trọng. Thứ nhất, khi ngôn ngữ đặc tả tác tử trở nên ý nghĩa hơn, sau đó thậm chí suy diễn off-line được thực hiện. Thứ hai, các hệ thống được tạo ra theo cách này là không có khả năng học, (chẳng hạn, chúng không có khả năng thích nghi chương trình của chúng ở thời gian thi hành). Cuối cùng như với các hướng tiếp cận thực hiện trực tiếp, các khung đặc tả tác tử có khuynh hướng không có thông dịch tính toán cụ thể, dẫn đến không thể tổng hợp.

3.4.2.4 Sự xác minh

Khi phát triển một hệ thống cụ thể, ta cần phải chỉ ra rằng hệ thống này là đúng đắn với những gì đã có trong đặc tả ban đầu. Quá trình này được gọi là sự xác minh, và nó đặc biệt quan trọng nếu ta đã giới thiệu một cách không chính thức vào quá trình phát triển. Chúng ta có thể chia các hướng tiếp cận theo sự xác minh của các hệ thống thành hai lớp: (1) axiomatic; và (2) semantic (kiểm tra mô hình).

3.4.3 Các hướng tiếp cận tiêu đề (axiomatic)

Các hướng tiếp cận tiêu đề cho việc xác minh chương trình đã đưa vào xu hướng chính của khoa học máy tính. Việc xác minh tiên đề yêu cầu ta có thể nắm được chương trình cụ thể của mình và từ chương trình đó tìm ra được một cách ngữ nghĩa một lý thuyết logic mà thể hiện hành vi của chương trình. Ta gọi nó là lý thuyết chương trình. Nếu như lý thuyết chương trình được biểu diễn trong một ngôn ngữ logic giống như của đặc tả ban đầu thì việc xác minh giảm tới một bài toán xác minh: chỉ ra rằng đặc tả là một định lý của lý thuyết chương trình.

Sự phát triển của một lý thuyết chương trình được tạo ra khả thi bằng cách tiên đề hoá ngôn ngữ lập trình trong đó hệ thống được thực hiện. Ví dụ, logic Hoare cho chúng ta ít hoặc nhiều hơn một tiên đề cho mỗi loại câu lệnh trong một ngôn ngữ giống Pascal đơn giản. Từ sự tiên đề hoá đã cho, lý thuyết chương trình có thể được đưa ra từ nguyên bản chương trình theo cách ngữ nghĩa.

Có thể nghiên cứu liên quan nhất từ xu hướng chung của khoa học máy tính là đặc tả và xác minh các hệ thống phản xạ sử dụng logic thời gian, được mở đường

bơi pnueli, Manna và các đồng nghiệp. Ý tưởng là các việc tính toán của hệ thống phản xạ là các chuỗi không giới hạn mà tương đương với các mô hình cho logic thời gian tuyến tính. Logic thời gian có thể được sử dụng để phát triển một đặc tả hệ thống và để tiên đề hoá một ngôn ngữ lập trình. Sự tiên đề hoá này sau đó có thể được sử dụng để đưa ra một cách ngữ nghĩa lý thuyết của một chương trình từ nguyên bản chương trình. Các đặc tả và lý thuyết chương trình sau đó sẽ được mã hoá trong logic thời gian, và việc xác minh sau đó trở thành một bài toán chứng minh trong logic thời gian.

3.4.3.1 Sự tiên đề hoá hai ngôn ngữ đa tác tử

Một nghiên cứu tương đối nhỏ đã được thực hiện trong nhóm các hệ thống dựa tác tử trên sự tiên đề hoá các môi trường đa tác tử. Ta sẽ xem lại chi một hướng tiếp cận. Một tiếp cận tiên đề về việc xác minh của hệ thống đa tác tử được đề xuất. Về bản chất, ý tưởng là để sử dụng một logic belief thời gian để tiên đề hoá các thuộc tính của hai ngôn ngữ lập trình đa tác tử. Từ một sự tiên đề hoá đã biết, một lý thuyết chương trình thể hiện các thuộc tính của hệ thống có thể được đưa ra một cách hệ thống theo cách đã chỉ ra ở trên. Một logic belief thời gian đã được sử dụng vì hai lý do.

Trước hết, một thành phần thời gian được yêu cầu vì như đã quan sát ở trên, chúng ta cần nắm bắt hành vi đang xảy ra của hệ thống đa tác tử. Một thành phần belief đã được sử dụng bởi vì các tác tử chúng ta muốn xác minh là mỗi hệ thống trí tuệ nhân tạo (AI) tượng trưng trong quyền của chính nó. Điều đó có nghĩa là mỗi tác tử là một hệ thống suy diễn tượng trưng bao gồm một sự biểu diễn môi trường của nó và hành vi mong ước. Một thành phần belief về mặt logic cho phép nắm bắt các biểu diễn tượng trưng thể hiện trong mỗi tác tử. Lưu ý rằng hướng tiếp cận này dựa vào thao tác của các tác tử là đơn giản một cách thoả đáng rằng các thuộc tính của chúng có thể được tiên đề hoá về mặt logic. Với các tác tử phức tạp hơn, một sự tiên đề hoá là không quá phức tạp. Hơn nữa, việc nắm bắt ngữ nghĩa của việc thực hiện đồng thời của các tác tử là không dễ dàng.

3.4.3.2 Các hướng tiếp cận ngữ nghĩa: kiểm tra mô hình

Sau cùng, sự xác minh tiên đề giảm chỉ còn là một bài toán chứng minh. Cách tiếp cận tiên đề để xác minh vốn đã bị giới hạn bởi sự phức tạp của bài toán chứng minh này. Các sự chứng minh là đủ phức tạp, thậm chí trong logic cổ điển; các kết nối hình thức hoặc thời gian đến một logic làm cho bài toán phức tạp hơn đáng kể. Vì lý do này, các hướng tiếp cận hiệu quả cho việc xác minh đã được yêu cầu. Một hướng tiếp cận thành công đặc biệt là kiểm tra mô hình (model checking). Như tên gọi đã gợi ý, trong khi tiếp cận tiên đề thường là dựa vào sự chứng minh cú pháp, cách tiếp cận kiểm tra mô hình được dựa trên ngữ nghĩa của ngôn ngữ đặc tả.

Về mặt lý thuyết, bài toán kiểm tra mô hình hoàn toàn đơn giản: một công thức ϕ của ngôn ngữ L được biết trước, mô hình của L, xác định cho dù ϕ có hợp lệ trong M hay không, tức là $M \models \phi$. Sự xác minh dựa trên việc kiểm tra mô hình đã được nghiên cứu có liên quan tới logic thời gian. Kỹ thuật này lần nữa được dựa vào mối quan hệ gần gũi giữa các mô hình logic thời gian và các máy trạng thái giới hạn. Giả sử rằng ϕ là đặc tả cho hệ thống nào đó và Π là một chương trình thực hiện ϕ . Sau đó, để xác minh xem Π có phải là thực hiện đúng đắn ϕ hay không, ta lựa chọn Π rồi từ đó sinh ra một mô hình M_Π tương ứng với Π , với ý nghĩa rằng M_Π mã hóa tất cả các tính toán có thể của Π . Để xác định xem $M_\Pi \models \phi$ hay không, tức là đặc tả công thức ϕ là hợp lệ trong M_Π ; chương trình Π thỏa mãn đặc tả ϕ chỉ trong trường hợp câu trả lời là “yes”. Thuận lợi chính của kiểm tra mô hình qua sự xác minh tiên đề là sự phức tạp: kiểm tra mô hình sử dụng việc phân nhánh thời gian logic thời gian CTL có thể được thực hiện trong thời gian đa thức, ngược lại bài toán chứng minh cho hầu hết các logic hình thức là hoàn toàn thích hợp.

Kiểm tra mô hình các hệ thống BDI

Rao và Georgeff đưa ra một thuật toán cho việc kiểm tra mô hình các hệ thống tác tử. Chính xác hơn họ đưa ra một thuật toán cho việc lựa chọn một mô hình logic cho ngôn ngữ đặc tả tác tử BDI của họ, một thể thức của ngôn ngữ và việc quyết định thể thức đó có hiệu lực lên mô hình hay không. Kỹ thuật gần như là dựa trên giải thuật kiểm tra mô hình cho các logic hình thức chuẩn. Họ chỉ ra rằng mặc dù

kết luận của ba thể thức đặc biệt (beliefs, desires, intentions) vào việc phân nhánh khung thời gian, thuật toán vẫn hoàn toàn hiệu quả khi chạy trong thời gian đa thức. Bởi vậy bước thứ hai của quá trình kiểm tra hai giai đoạn được mô tả ở trên có thể vẫn được thực hiện một cách hiệu quả. Tuy nhiên, cách mà bước đầu tiên thực hiện cho logic BDI là vẫn không rõ ràng. Ở đâu thì mô hình logic mô tả đặc điểm một tác tử thực tế có thể được xuất phát từ một chương trình π bất kỳ, như ở trong một xu hướng chủ đạo của khoa học máy tính? Để thực hiện điều này, ta cần lựa chọn một chương trình được thực hiện bởi Pascal và từ đó sinh ra các mối quan hệ có thể truy cập belief, desire, intention mà được sử dụng để đưa ra một ngữ nghĩa cho thành phần BDI của logic. Như ta đã lưu ý ở trên, bởi vì không có một mối quan hệ rõ ràng giữa logic BDI và các mô hình tính toán cụ thể được sử dụng để thực hiện các tác tử, cách mà một mô hình có thể được đề xuất cũng là không rõ ràng.

3.5 Phương pháp luận hướng tác tử

Công nghệ phần mềm hướng tác tử đã trở thành một lĩnh vực nóng hổi của nghiên cứu trong những năm gần đây. Đã rất nhiều phương pháp luận được đưa ra. Trong phần này chúng ta sẽ xem xét một số phương pháp phân tích thiết kế hướng tác tử, và xem xem chúng thỏa mãn những yêu cầu của phân tích thiết kế hướng tác tử như thế nào.

3.5.1 Phương pháp Prometheus

Pha phân tích của phương pháp Prometheus bao gồm:

- Xác định môi trường của hệ thống trong những điều kiện của: nhận thức (nhưng thông tin đi từ môi trường), những hành động (những hành động mà một tác tử dùng để tác động lên môi trường của nó), và dữ liệu bên ngoài.
- Xác định những mục đích, những chức năng và kịch bản use-case của hệ thống
- Những mục đích xác định những gì mà hệ thống sẽ làm.
- Những chức năng tương tự như vai trò trong những phương pháp khác (Gaia). Việc định nghĩa chức năng để hoàn thành những mục đích đã được xác định.

- Kịch bản use-case miêu tả hệ thống đang được sử dụng. Bao gồm cả tri thức gì được thừa nhận, những thông điệp nào được gửi đi, và những hành động nào được thực hiện bởi tác tử.

Những chức năng Prometheus được miêu tả dưới đây:

- Tên
- Những miêu tả ngôn ngữ tự nhiên ngắn gọn
- Danh sách những hành động
- Danh sách những tri thức thích hợp
- Dữ liệu được sử dụng.
- Dữ liệu được tạo ra
- Những tương tác với những chức năng khác.

Miêu tả của “những tương tác với những chức năng khác” xác định thông điệp nào mà vai trò gửi và nhận từ những vai trò khác. Điều này ràng buộc cách mà vai trò tương tác với những tác tử khác, và ràng buộc về việc một vai trò tương tác với những vai trò khác. Tuy nhiên, những chức năng của Prometheus lại không xác định chi tiết cách mà những chức năng được hoàn thành như thế nào. Bởi vậy, trong khi pha phân tích của Prometheus bắt đầu xác định cách mà những chức năng liên hệ tới những chức năng khác, thì những chức năng Prometheus lại có xu hướng ít thực hiện được hơn là vai trò trong Gaia. Nó nắm bắt những mục tiêu của hệ thống, tri thức về phạm vi ứng dụng, những yêu cầu trên môi trường, và những yêu cầu trên hệ thống máy tính. Những mô hình được tạo ra miêu tả những yêu cầu của hệ thống và tránh gồm quá nhiều những chi tiết về cách mà hệ thống được thực hiện ra sao.

3.5.2 Phương pháp Tropos

Phương pháp tropos được thiết kế bao gồm tất cả các pha trong phát triển phần mềm. Nó bao gồm những pha cho yêu cầu sớm, yêu cầu muộn, thiết kế kiến trúc, thiết kế chi tiết và thực hiện. Những pha phân tích phần mềm là những pha yêu cầu sớm và yêu cầu muộn. Những khái niệm và ký hiệu tương tự được sử dụng trong cả hai pha. Những khái niệm trong phương pháp.

-**Mục đích:**Biểu diễn những quan tâm chiến lược của người thực hiện. Tropos miêu tả những mục đích cứng và mềm. Những mục đích cứng miêu tả chức năng mà hệ thống sẽ hoàn thành. Những mục đích mềm miêu tả những yêu cầu phi chức năng như là bảo mật, sự thực hiện và khả năng bảo trì.

-**Ràng buộc:**Mối quan hệ ràng buộc giữa những người thực hiện miêu tả những trường hợp ở đó một người thực hiện phụ thuộc vào người khác để đạt được một mục đích thực hiện một kế hoạch, hoặc chuyển giao tài nguyên.

-**Kế hoạch:**Là một cách đặc biệt để đạt được một mục đích.

-**Tài nguyên:**Những tài nguyên vật lý hay dữ liệu mà một người thực hiện có thể cần hoặc người khác cung cấp.

-**Năng lực:**Khả năng của một con người thực hiện để định nghĩa, chọn lựa và thực hiện một kế hoạch để hoàn thành mục đích.

-**Tri thức:**Biểu diễn những tri thức của người thực hiện.

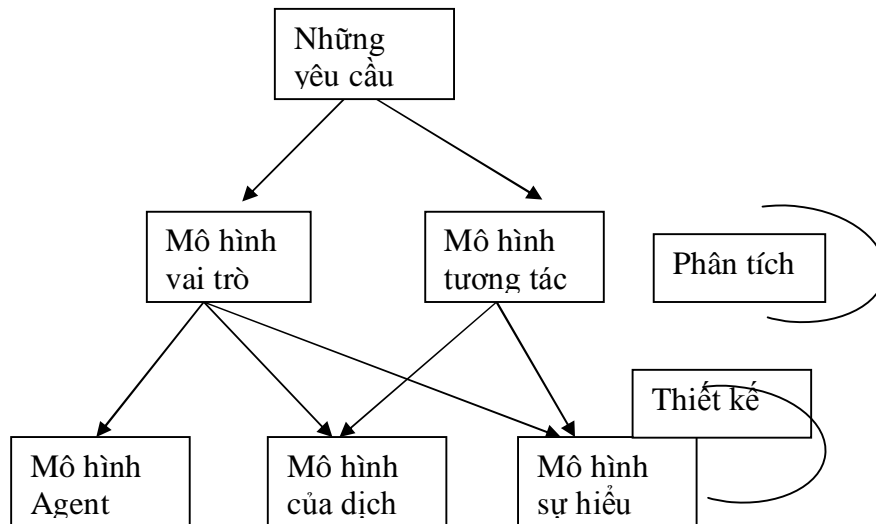
Trong pha *yêu cầu sớm*, chúng ta mô hình hoá và phân tích những mục đích và những ý định của nhà đầu tư. Thông qua phân tích, phương pháp Tropos xây dựng những mô hình yêu cầu chức năng và phi chức năng. Những mô hình này bao gồm thực hiện, mục đích của chúng, và những ràng buộc của chúng.

3.5.3 Phương pháp Gaia

Gaia mục đích là cho phép một nhà phân tích đi theo một cách hệ thống từ những yêu cầu đến một thiết kế mà đã được mô tả đầy đủ đến mức nó có thể thực hiện được một cách trực tiếp. Ghi nhớ rằng chúng ta xem xét những yêu cầu nắm bắt các pha như thể độc lập với các mô hình được sử dụng cho phân tích và thiết kế. Trong khi áp dụng Gaia, người phân tích đi từ trừu tượng đến dần khai niệm cụ thể. Mỗi bước tiến kế tiếp sẽ đưa ra xu hướng thực hiện lớn hơn, và giảm không gian những hệ thống có thể mà có thể được thực hiện để thoả mãn những yêu cầu ban đầu. Phân tích và thiết kế có thể được xem như là một tiến trình của việc phát triển những mô hình được chi tiết hoá hơn của hệ thống được xây dựng. Những mô hình chính được sử dụng trong phương pháp Gaia được tổng kết lại trong hình 5.3, Gaia

đã mượn một số những kỹ thuật và ký hiệu của phân tích và thiết kế hướng đối tượng.

Tuy nhiên, đây không phải là một cố gắng vô ích để áp dụng những phương pháp như vậy cho phát triển hướng tác tử. Hơn nữa nó cung cấp một tập hợp những tác tử riêng biệt của những khái niệm mà thông qua đó một kỹ sư phần mềm có thể hiểu và mô hình lên được một hệ thống phức tạp. Trong trường hợp cụ thể, Gaia khuyến khích những nhà phát triển nghĩ đến việc xây dựng những hệ thống dựa trên tác tử như là một công việc thiết kế tổ chức. Những khái niệm chính của Gaia có thể được chia thành hai loại: trừu tượng và cụ thể; những khái niệm trừu tượng và cụ thể được đúc kết trong bảng 2. Những thực thể trừu tượng được sử dụng trong quá trình phân tích để mô hình hoá hệ thống, nhưng lại không cần thiết phải được sự thực hiện trực tiếp trong hệ thống. Những thực thể cụ thể, ngược lại được sử dụng trong quá trình thiết kế và sẽ có những bản sao trong khi thực thi hệ thống.

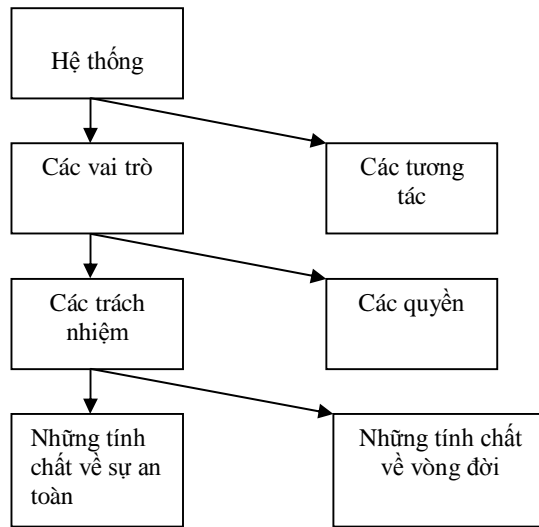


Hình 3.3 *Mối quan hệ giữa những mô hình Gaia*

Những khái niệm trừu tượng	Những khái niệm cụ thể
Vai trò	Các kiểu Agent
Quyền	Các dịch vụ
Trách nhiệm	Sự hiểu biết

Giao thức	
Hoạt động	
Những tính chất về dòng đời	
Những tính chất về sự an toàn	

Bảng 3.1 Những khái niệm trừu tượng và cụ thể trong Gaia



Hình 3.4 Những khái niệm trừu tượng

3.6 Một số ví dụ về ứng dụng công nghệ tác tử

3.6.1 Ví dụ về cách thức di chuyển của robot áp dụng tác tử

Một tác tử Robot có nhiệm vụ hút bụi trong một căn phòng. Tác tử được trang bị một máy hút bụi và một cảm biến cho phép phát hiện tại vị trí robot đang đứng có bụi hay không; máy hút bụi chỉ có khả năng hút bụi tại vị trí hiện thời của robot. Robot có khả năng di chuyển về phía trước và có khả năng quay sang phải 90⁰ (để đơn giản ta không xét khả năng quay trái của tác tử). Bằng cách quay phải, robot có thể thay đổi hướng chính diện của mình bao gồm *đông, tây, nam, bắc*. Để đơn giản hóa việc định vị, căn phòng hút bụi được chia thành các ô vuông, chuyển động của robot được chia thành từng đơn vị, mỗi đơn vị chuyển động cho phép

robot di chuyển sang ô bên cạnh, nếu ô đó không phải là tường. Ví dụ này kích thước phòng là 3×3 ô như hình vẽ. Ta giả thiết vị trí xuất phát là ô(0,0) và hướng về phía bắc.

Như vậy cảm nhận của robot bao gồm: vị trí hiện tại có bụi hay không; hành động của tác tử bao gồm: *tiến lên* (theo hướng chính diện), *quay* (phải) và *hút* (bụi). Nhiệm vụ của robot là di chuyển trong phòng và hút bụi từ những ô có bụi.

Để lưu giữ thông tin môi trường và suy diễn, tác tử duy trì trạng thái bên trong dưới dạng biểu thức logic. Thông tin về môi trường được biểu diễn bởi những vị từ sau:

vị_trí (x,y) robot đang ở ô (x,y)

có_bụi (x,y) tại ô (x,y) có bụi

hướng (d) hướng chính diện của tác tử là d

Hành vi của tác tử được quy định bởi luật suy diễn:

$vị_trí(x,y) \wedge có_bụi(x,y) \rightarrow thực_hiện(hút)$ (2.1)

Trong ví dụ đang xét, hành động này có mức ưu tiên cao nhất, tức là khi robot có thể hoặc hút hoặc di chuyển, tác tử phải ưu tiên hút trước và di chuyển sau. Đây là đảm bảo tác tử sẽ hút bụi chứ không di chuyển đơn thuần. Để đơn giản ta quy định sẵn quỹ đạo chuyển động của tác tử là (0,0), (0,1), (0,2), (1,2), (1,1), (1,0), (2,0), (2,1), (2,2). Chuyển động của robot được xác định bởi một số luật đơn giản:

$Vị_trí(0,0) \wedge Hướng(bắc) \wedge \neg Có_bụi(0,0) \rightarrow Thực_hiện(tiến_lên)$ (2.2)

$Vị_trí(0,1) \wedge Hướng(bắc) \wedge \neg Có_bụi(0,1) \rightarrow Thực_hiện(tiến_lên)$ (2.2)

$Vị_trí(0,2) \wedge Hướng(bắc) \wedge \neg Có_bụi(0,2) \rightarrow Thực_hiện(quay)$ (2.4)

$Vị_trí(0,2) \wedge Hướng(đông) \rightarrow Thực_hiện(tiến_lên)$ (2.5)

.....

Các luật cho phép tác tử di chuyển tiếp đến ô (2,2) được xây dựng tương tự. Biểu thức $\neg Có_bụi(x,y)$ được thêm vào các luật trên đảm bảo tác tử chỉ chuyển động khỏi một ô khi chắc chắn ô đó không có bụi, nhờ vậy thao tác hút bụi luôn có mức ưu tiên cao hơn thao tác di chuyển.

0,2	1,2	2,2
0,1	1,1	2,1
0,0	1,0	1,1

Hình 3.5 Tác tử robot hút bụi trong ô 3×3

Để trạng thái phản ánh đúng thông tin về môi trường, ta cần xác định hàm *Cập_nhật*. Hàm cập nhật thông tin do cảm biến đưa vào (một trong hai giá trị *có_bụi* hoặc *không_bụi*) để thay đổi giá trị vị từ *Có_bụi(x,y)*. Ngoài ra dựa trên hành động vừa thực hiện, hàm cập nhật thay đổi sự kiện về vị trí và hướng chính diện chứa trong trạng thái tác tử. Hàm cập nhật có thể cho như sau:

Có_bụi(x,y) nếu giá trị cảm biến tại (x,y) là *có_bụi*

– *Có_bụi(x,y)* nếu ngược lại

Kết_quả(*Vị_trí(0,0)*, Thực_hiện(*tiến_lên*))=*Vị_trí(0,0)*

Kết_quả(*Vị_trí(0,2)*, Thực_hiện(*quay*))=*Vị_trí(1,2)*

.....

Như vậy ta đã hoàn thành mô hình tác tử hút bụi bao gồm tri thức, hàm *Cập_nhật* và hàm *Hành_động* để di chuyển robot với yêu cầu của bài toán.

3.6.2 Ví dụ về đấu thầu sử dụng tác tử

Đấu thầu được sử dụng trong nhiều ứng dụng khác nhau. Một trong những ứng dụng đó là thiết lập mạng cảm biến phân tán. Mạng cảm biến phân tán bao gồm một số nút được bố trí phân tán trong một khu vực rộng với nhiệm vụ theo dõi sự di chuyển của xe cộ trong khu vực đó. Mỗi nút được đặt tại một vị trí cố định và không biết trước vị trí của nút khác. Các nút được chia làm hai loại: loại 1 (kí hiệu S) được trang bị cảm biến âm thanh và có khả năng phát hiện xe trong phạm vi hoạt động của mình; loại thứ 2 kí hiệu P chỉ có khả năng xử lý thông tin. Ngoài ra còn có một nút M có nhiệm vụ tổng hợp kết quả cảm biến và gửi cho người ra quyết định. Tổ chức hệ thống cảm biến phân tán được minh họa như hình vẽ(). Vòng tròn quanh các nút là phạm vi hoạt động của cảm biến. Đường thẳng chạy ngang qua là

quỹ đạo chuyển động của một xe. Do khu vực cần theo dõi rộng hơn phạm vi hoạt động của từng nút, không nút nào có khả năng theo dõi toàn bộ khu vực một cách độc lập. Nếu coi mỗi nút là một tác tử thì đây là một ví dụ hệ đa tác tử trong đó trong từng tác tử không đủ khả năng giải quyết công việc một mình.

Nhiệm vụ của hệ thống là tổng hợp một số lượng lớn dữ liệu cảm biến được, lọc bớt dữ liệu trùng lặp và biến đổi thành dạng thích hợp cho người sử dụng. Dữ liệu và công việc được phân cấp như hình vẽ. Ở mức thấp nhất, dữ liệu có dạng tín hiệu thô. Các tín hiệu liên quan đến nhau tạo thành các nhóm (mức thứ hai từ dưới lên). Tín hiệu ở mức cao hơn tương ứng với xe. Mỗi xe được mô tả bởi một số nhóm tín hiệu liên quan và bao gồm các đặc điểm như: vị trí, tốc độ, hướng chuyển động, loại xe (do động cơ từng loại xe có tần số âm thanh phát ra khác nhau nên việc phân tích tín hiệu âm cho phép xác định âm thanh đó là thuộc loại xe nào). Vị trí xe được xác định bằng cách phân tích tín hiệu thu được từ một số cảm biến lân cận, tốc độ và hướng chuyển động có thể xác định bằng cách theo dõi vị trí xe theo thời gian. Sơ đồ khu vực là mức tiếp theo trong phân cấp dữ liệu. Mức này bao gồm thông tin về chuyển động của xe trong một khu vực nhỏ. Thông tin từ tất cả các khu vực nhỏ tạo thành sơ đồ toàn khu vực và được xử lý bởi nút M.

Phân cấp công việc được tạo thành từ phân cấp dữ liệu. Tác tử tại nút M quản lý một số nhà thầu khu vực. Mỗi nhà thầu loại này chịu trách nhiệm lập sơ đồ xe trong khu vực của mình. Mỗi nhà thầu khu vực lại quản lý một số nhà thầu tín hiệu và nhà thầu xe. Nhà thầu tín hiệu tổng hợp tín hiệu từ các nút cảm biến và cung cấp cho nhà thầu khu vực trong khi nhà thầu xe chịu trách nhiệm theo dõi và cung cấp thông tin về xe của mình. Mỗi nhà thầu xe, để thực hiện nhiệm vụ của mình, lại quản lý ba nhà thầu làm các công việc phân loại, định vị và theo dõi.

Hoạt động của hệ thống được chia thành hai giai đoạn: khởi tạo và thực hiện. Tác tử M chịu trách nhiệm khởi tạo hệ thống và tổng hợp sơ đồ toàn khu vực. Khởi đầu, M chọn trong số nút P một số nhà thầu khu vực và phân chia khu vực cho các nhà thầu này. Do đặc điểm hệ thống, lúc đầu, M chỉ biết tên các nút P mà không biết vị trí. M gửi thông báo mời thầu cho các nút P. Thông báo chứa yêu cầu về khả

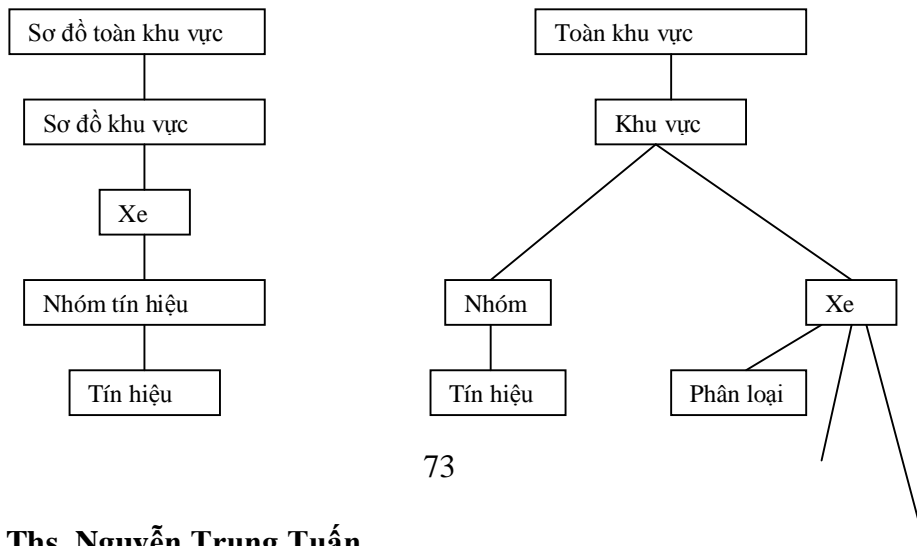
năng mới thầu-có vị trí phù hợp-và yêu cầu đối với đề nghị bỏ thầu-ở đây là mô tả khu vực sẽ giám sát. Cần nhắc lại yêu cầu với đề nghị bỏ thầu mô tả thông tin cần có trong đề nghị để tác tử quản lý có thể lựa chọn trong số đề nghị nhận được.

Sau khi nhận được thông báo, tác tử P gửi lại đề nghị bỏ thầu có chứa thông tin về vị trí và khu vực giám sát dự kiến của mình. Trên cơ sở thông tin này, M phân chia khu vực thành khu vực con lựa chọn một số nhà thầu khu vực. Thông báo được gửi cho những nút P thắng thầu.

Sau khi được giao nhiệm vụ, nhà thầu khu vực tổng hợp dữ liệu về xe để hình thành sơ đồ khu vực của mình. Để có được dữ liệu, nhà thầu khu vực cần thu thập từ những nút khác. Do không biết nút nào có khả năng cung cấp các dữ liệu này, nhà thầu khu vực thông báo về nhiệm vụ để các nút khác. Thông báo bao gồm thông tin về khu vực cần giám sát. Đề nghị bỏ thầu bao gồm vị trí các nút trong khu vực. Các nút thắng thầu trở thành nhà thầu nhóm tín hiệu và nhà thầu xe.

Quá trình chia sẻ công việc được tiếp diễn liên tục cho đến khi tìm được nhà thầu ở mức thấp nhất: mức tín hiệu, định vị, phân loại và theo dõi.

Sử dụng giao thức đấu thầu trong ví dụ trên cho phép chia sẻ công việc một cách mềm dẻo cho các ứng dụng trong đó cấu trúc nhiệm vụ không được biết từ trước và có thể thay đổi trong quá trình hoạt động. Các phần việc còn quá lớn có thể phân rã thành phần việc nhỏ hơn và phân chia tiếp tới những tác tử thích hợp. Mỗi tác tử đều có thể đóng vai trò người giao việc hoặc người thực hiện.



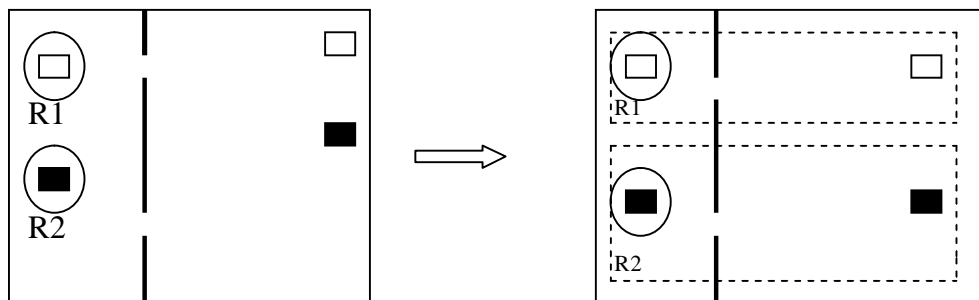


Hình 3.6 Phân cấp dữ liệu (a) và phân cấp công việc

3.6.2 Ví dụ về việc lập kế hoạch phân tán của tác tử

Cho hai robot với nhiệm vụ di chuyển giữa hai phòng như trên hình vẽ. Để di chuyển đồ vật, robot phải chuyển động liên tục từ phòng này sang phòng khác. Đường đi ngắn nhất đối với cả hai robot đều đi qua cùng một cửa (cửa trên) do đó có thể xung đột với nhau. Sau khi trao đổi kế hoạch, hai robot phát hiện ra mâu thuẫn này. Robot có thể giải quyết mâu thuẫn ở mức trừu tượng cao bằng cách phân chia cố định cho mỗi robot một không gian riêng (cửa riêng). Cách này dẫn tới việc phân chia không gian như hình vẽ. Mặc dù đơn giản, cách giải quyết như vậy là không hiệu quả do R2 luôn sử dụng đoạn đường dài hơn.

Nếu xem xét chuyển động của mình chi tiết hơn, robot có thể tìm ra giải pháp khác. Giải pháp ở đây là xem xét kết hợp nhu cầu sử dụng không gian với thời gian. R2 có thể đợi cho đến khi R1 sử dụng xong cửa trên trước khi sử dụng cửa này. Ở đây robot có thể thay đổi ở những mức chi tiết khác nhau. Ở mức cao nhất, R2 đợi cho R1 hoàn thành công việc trước khi bắt đầu sử dụng cửa trên. Ở mức trung gian, R2 bắt đầu chuyển động khi R1 rời khỏi cửa. Ở mức chi tiết nhất, R2 có thể chuyển động cùng R1 và chỉ dừng lại ở gần cửa để nhường R1 đi qua trước.



Hình 3.7 Giải pháp mâu thuẫn nhờ phân chia không gian

Kết luận và đánh giá

Qua tìm hiểu đã giúp cho ta thấy được tổng quan về tác tử là gì? Đặc điểm và thành phần của tác tử, sự phối hợp của tác tử trong hệ đa tác tử để xử lý các vấn đề trong thực tế. Tác tử rõ ràng đã giải quyết được những vấn đề phức tạp mà các công nghệ trước chưa giải quyết tốt hay chưa giải quyết được. Tác tử có những ưu điểm vượt trội so với các công nghệ khác đó là tính thông minh, cũng như khả năng tự chủ hành động mà không cần có sự can thiệp trực tiếp của con người, tự động hành động để đạt được mục tiêu mà không cần phải lập trình trước đó- thể hiện khả năng có thể thích nghi với mọi điều kiện môi trường và tác động lại vào môi trường có lợi cho con người.

Dù công nghệ tác tử còn mới mẻ đối với chúng ta nhưng những gì mà công nghệ này mang trong nó sẽ được nghiên cứu nhiều và phát triển trong tương lai. Tác tử đã có những ứng dụng để giải quyết các bài toán trong thương mại điện tử, trong quản lý sản xuất,... Trong thời đại công nghệ thông tin thì những ứng dụng của tác tử trong xã hội là rất cần thiết để giảm sức lao động của con người đồng thời nâng cao được hiệu quả lao động. Từ những lý do trên có thể đánh giá công nghệ tác tử là công nghệ hữu ích và là giải pháp để giải quyết nhiều vấn đề trong xã hội và triển vọng của công nghệ này là rất sáng sủa.

Tài Liệu Tham Khảo

- Tác tử-Công nghệ phần mềm hướng tác tử
của Lê Tấn Hùng, Từ Minh Phương, Huỳnh Quyết Thắng
- Công nghệ Agent trong quản lý mạng internet
của Biện Văn Quang (quantrimang.com)
- www.bcs.Org
- www.W3.org
- www.dse.su.se