|  |
| --- |
| ĐẠI HỌC QUỐC GIA HÀ NỘI  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**    **Nhóm 7:** Lê Văn Hùng  Vương Thị Mỵ  Phan Văn Nam  Bùi Thị Mai  Nguyễn Thị Thùy Dương  **TÌM HIỂU HỆ ĐA TÁC TỬ VỚI JADE VÀ ÁP DỤNG MÔ HÌNH ĐA TÁC TỬ VỚI BÀI TOÁN NHÀ THÔNG MINH**  **Hà Nội - 2016** |

**MỤC LỤC**

[CHƯƠNG I: CƠ SỞ LÍ THUYẾT 2](#_Toc443933760)

[I.1. Tác tử và hệ đa tác tử 2](#_Toc443933761)

[I.1.1. Giới thiệu về tác tử và hệ đa tác tử 2](#_Toc443933762)

[I.1.2. Định nghĩa về tác tử 2](#_Toc443933763)

[I.1.3. Các đặc điểm tính chất của tác tử 2](#_Toc443933764)

[I.1.4. Các loại tác tử 3](#_Toc443933765)

[I.2. Hệ đa tác tử 3](#_Toc443933766)

[I.2.1. Khái niệm 3](#_Toc443933767)

[I.2.2. Sự phối hợp của các tác tử 3](#_Toc443933768)

[I.2.3. Ưu điểm và nhược điểm 4](#_Toc443933769)

[I.2.4. Ứng dụng 4](#_Toc443933770)

[I.3. Kiến trúc JADE 4](#_Toc443933771)

[I.3.1. Các thành phần JADE 4](#_Toc443933772)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ PHẦN MỀM 7](#_Toc443933773)

[CHƯƠNG III: CÀI ĐẶT VÀ TRIỂN KHAI 8](#_Toc443933774)

[III.1 Cài đặt 8](#_Toc443933775)

**DANH SÁCH CÁC TỪ VIẾT TẮT**

JADE **J**ava **a**gent **de**velopment Framework

FIPA **F**oundation for **I**ntelligent **P**hysical **A**gents

**DANH MỤC HÌNH ẢNH**

[Hình 1: Khung chứa và các nền tảng 5](#_Toc443933781)

[Hình 2: Mô tả cách thêm các thư viện JADE để cấu hình một project 8](#_Toc443933782)

[Hình 3: Mô tả cách cấu hình các tham số trong JADE để chạy một chương trình 9](#_Toc443933783)

# ĐẶT VẤN ĐỀ

# CHƯƠNG I: CƠ SỞ LÍ THUYẾT

## I.1. Tác tử và hệ đa tác tử

### I.1.1. Giới thiệu về tác tử và hệ đa tác tử

Chương này trước hết giới thiệu các khái niệm về tác tử, tổng quan các công nghệ tác tử, kiến trúc tác tử, các ngôn ngữ lập trình và các công cụ phát triển. Tiếp theo sẽ mô tả các đặc tả của FIPA- tập các tiêu chuẩn phổ biến nhất và được chấp nhận rộng rãi cho phát triển các nền tảng và ứng dụng đa tác tử. JADE là một nền tảng tuân theo các đặc tả FIPA và hơn nữa nó còn mở rộng mô hình FIPA trong một số lĩnh vực như tác tử cho thiết bị di động, tác tử cho dịch vụ web.

### I.1.2. Định nghĩa về tác tử

Thuật ngữ tác tử (Agent) hay tác tử phần mềm đã được sử dụng rộng rãi và xuất hiện trong nhiều lĩnh vực nghiên cứu như trí tuệ nhân tạo, cơ sở dữ liệu, các tài liệu về hệ điều hành và mạng máy tính.

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

Trong đó:

Hệ thống tính toán có thể là phần cứng, phần mềm, hoặc cả phần cứng lẫn phần mềm.

Khả năng cảm nhận môi trường: tác tử nhận thông tin từ môi trường qua các cơ quan cảm nhận.

Tác động vào môi trường: tác tử tác động vào môi trường bằng các cơ quan tác động.

### I.1.3. Các đặc điểm tính chất của tác tử

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.

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 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.

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.

### I.1.4. Các loại tác tử

Có bốn loại tác tử:

Tác tử phản xạ đơn giản: hành động theo một quy tác có điều kiện phù hợp với trạng thái hiện thời.

Tác tử phản xạ dựa trên mô hình: sử dụng một mô hình nội bộ để giám sát trạng thái hi thái hiện tại của môi trường.

Tác tử dựa trên mục tiêu:

Tác tử dựa trên lợi ích:

## I.2. Hệ đa tác tử

### I.2.1. Khái niệm

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.

Một hệ đa tác tử là một tập hợp các tác tử có mục đích riêng, miền tri thức riêng nhưng có thể tương tác với nhau để hoàn thành mục tiêu chung tổng thể của hệ thống.

### I.2.2. Sự phối hợp của các tác tử

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.

Cần có sự phối hợp giữa các tác tử vì:

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 chyền ngang. Trường hợp thứ hai, 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 thoả 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.

### I.2.3. Ưu điểm và nhược điểm

Mỗi tác tử sẽ hành động như một chương trình thông minh, quản lý các tài nguyên trong phạm vi của mình, kết hợp với các tác tử khác để chia sẻ tài nguyên. Với cách tiếp cận này, hệ thống sẽ dễ dàng phát triển và mở rộng vì tác tử hoạt động và phát triển độc lập.

### I.2.4. Ứng dụng

Trong lĩnh vực công nghiệp: kiểm soát tiến trình (Jenning, 1994), chẩn đoán hệ thống (Albert, 2003), sản xuất (Parunak, 1987), dịch vụ vận tải (Neagu, 2006), và quản lý mạng (Greenwood, 2006).

Trong lĩnh vực quản lý thông tin: sử dụng để tìm kiếm và lọc thông tin.

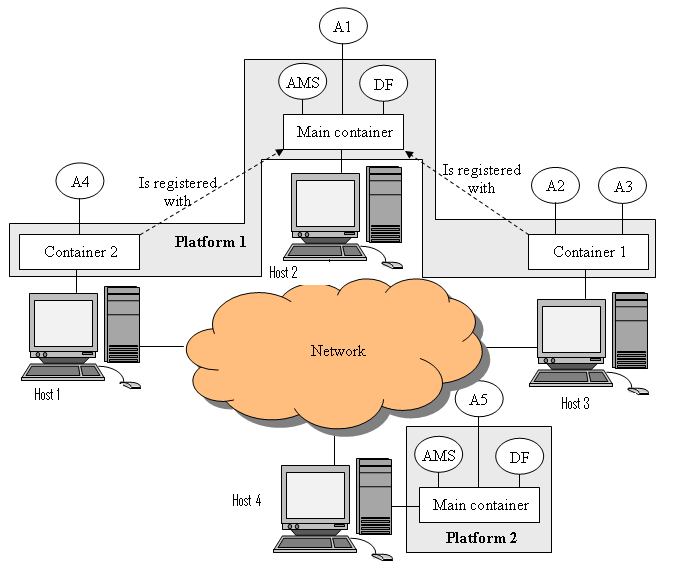
Trong lĩnh vực giao thông vận tải: Hệ thống OASIS là một hệ thống điều khiển không lưu phức tạp dự trên mô hình tác tử BDI, được triển khai và được sử dụng thành công tại sân bay Sydney ở Australia.

## I.3. Kiến trúc JADE

### I.3.1. Các thành phần JADE

Nền tảng JADE bao gồm các thành phần kiến trúc chính:

* Agent
* Container
* Platform
* Maincontainer
* AMS và DF



Hình 1: Khung chứa và các nền tảng

Agent là tập các thành phần agent và có tên duy nhất, mỗi Agent thực hiện nhiệm vụ và tương tác bằng cách trao đổi thông điệp. Containter: Khung chứa các Agent, có thể có hoặc không chứa các Agent, mỗi container có thể thực hiện trên các máy chủ khác nhau. Platform: mỗi flatform bao gồm một hoặc nhiều container. Main Container: chính là các container chứa các Agent, nhưng khác là phải là Containter đầu tiên để bắt đầu trong một flatform, để các containter khác đăng kí với nó thời gian khởi động.

AMS và DF: là hai agent đặc biệt chứa trong Main container.

**AMS**: đại diện quyền lực trong một flatform, là agent duy nhất có thể thực hiện hoạt động quản lý bắt đầu, tạm đừng hoặc kết thúc toàn bộ flatform, (agent bình thường có thể yêu cầu hoạt động như vậy tới AMS).

**DF**: cung cấp dịch vụ Page Yellow, nơi các agent có thể tạo ra các dịch vụ nó cung cấp và tìm các agent khác cung cấp các dịch vụ nó cần.

Nếu một Main container khác bắt đầu, điều này tạo ra một flatform mới.

Agent Communication:

Các Agent có thể giao tiếp thông suốt với nhau:

Trong cùng một container.

Trong các container khác nhau ( các container trong cùng host hoặc khác host, và trong cùng nên khác hoặc khác nền tảng).

Giao tiếp giữa các agent được dựa trên mô hình trao đổi thông điệp không đồng bộ.

Định dạng thông điệp được định nghĩa bởi ngôn ngữ ACL được xác định bởi FIPA ( tổ chức quốc tế ban hành một tập hợp các thông số kỹ thuật cho các agent tương thích). Một thông điệp ACL bao gồm:

The sender

The receiver(s)

The communicative act

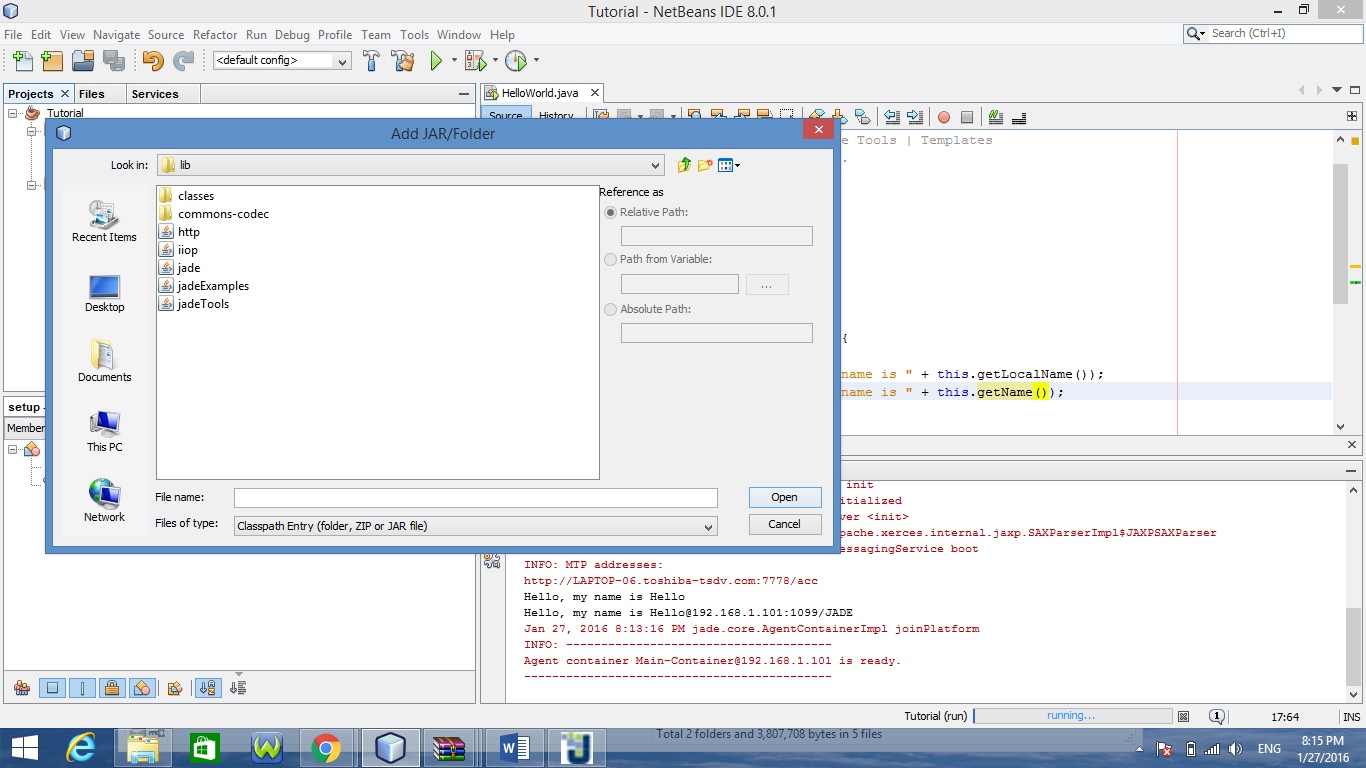
The content

# CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ PHẦN MỀM

# CHƯƠNG III: CÀI ĐẶT VÀ TRIỂN KHAI

## III.1 Cài đặt

Trước tiên, chúng ta tải framework JADE tại trang chủ sau: <http://jade.tilab.com/>. Khởi động Netbean và tạo một project mới, thêm các thư viện sau vào phần “Set Project Configuration” trong của project vừa tạo: http.jar, iiop.jar, jade.jar, jadeTools.jar, jadeExamples.jar, commons-codec-1.3.jar.



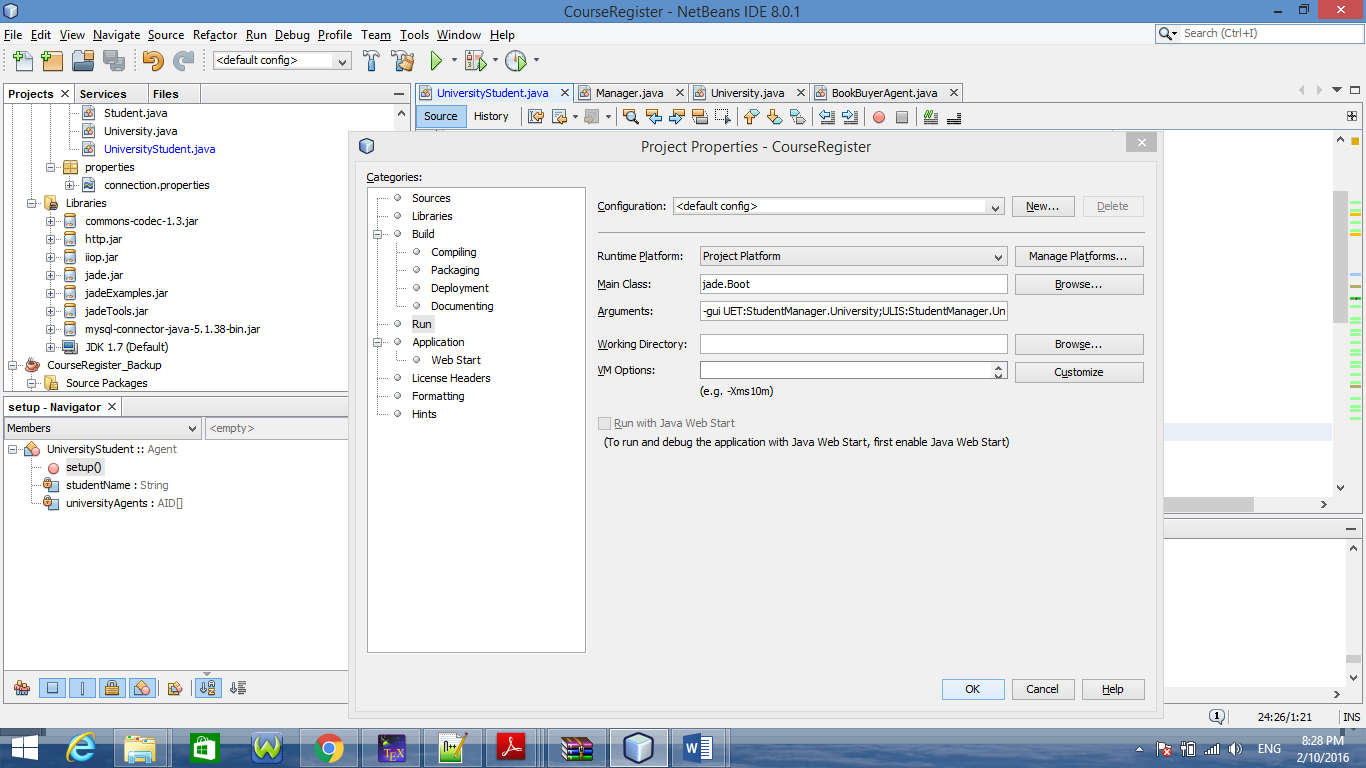
Hình 2: Mô tả cách thêm các thư viện JADE để cấu hình một project

Khi tạo một lớp tác tử mới phải kế thừa từ lớp Agent của package: jade.core.Agent, và cấu hình project của chúng ta với các tham số:

Main class: jade.Boot

Arguments: “-gui ‘Tên tác tử’:’Lớp thực thi tác tử’.

Ví dụ: “-guiUET:StudentManager.University;ulis:StudentManager.University”. Chúng ta viết chương trình thực thi trong phương thức setup().



Hình 3: Mô tả cách cấu hình các tham số trong JADE để chạy một chương trình

Jf dung để khai báo yello page

Các phương thức của Aggent:

getName(): xxxxxx@IP:port/JADE (VD: [Hello@192.168.1.101:1099/JADE](mailto:Hello@192.168.1.101:1099/JADE))

getLocalName(): Trả lại tên: xxxxx (VD: Hello)

Agent này sẽ sống mãi vì thế, khi show trạng thái của nó luôn là active. Để agent tự động xóa đi khi chạy xong thì ta sử dụng phương thức doDelete(), sau khi agent được chạy xong nó sẽ tự động xóa đi.

Truyền tham số cho các agent:

Vào customize -> Edit Arguments bằng cách thêm các tham số theo mẫu (Hung\_Le,Van\_Anh)

Gửi thông điệp giữa các Agents

Tạo 2 class Sender và Receiver, 2 class này đều extend lớp Agent.

**BEHAVIOUR**

Lập lịch và sự thực thi của các hành vi của các tác tử.

Một tác tử (agent) có thể thực hiện một vài hành vi một cách đồng thời. tuy nhiên, lịch trình hành vi thì không được ưu tiên, nhưng có sự hợp tác giữa các hành vi, với JADE có 4 kiểu hành vi

Hành vi thực hiện một lần (OneShot behavior)

Phương thức action() chỉ thực hiện duy nhất 1 lần. Vì thế phương thức done() mặc định là true. Hành vi này thuộc lớp jade.core.behaviours.OneShotBehaviour

Hành vi thực hiện liên tục (Cyclic Behaviour)

Đây là kiểu hành vi thực hiện mãi mãi, vì thế phương thức action() thực thi cùng một hành vi một cách liên tục, kết quả trả về của phương thức done() luôn luôn là false. Hành vi này thuộc lớp jade.core.behaviours.CyclicBehaviour

Hành vi được thực hiện trong điều kiện nhất định (WakerBehaviour)

Phương thức action() và done() được cài đặt trong phương thức onTick(). Phương thức onTick() được thực thi sau một khoảng thời gian nhất định.

Hành vi được thực hiện sau một khoảng thời gian nhất định (TickerBehaviour)

Phương thức action() và done() đã được cài đặt sẵn vì vậy phương thức onTick() được thực hiện định kì với một khoảng thời gian nhất định. Loại hành vi này được chạy mãi mãi trừ khi phương thức stop() của nó được thực thi.

Chạy ví dụ BookTrandding

Tạo package booktradding, dòng lệnh Arguments: -gui seller1:booktradding.BookSellerAgent;seller2:booktradding.BookSellerAgent

Cách thức hoạt động: 2 agents seller1 và seller2 sẽ đăng kí thông tin với yellow page những dịch vụ, pulic dịch vụ ấy, buyer sẽ gửi tín hiệu lên yellow page, nếu có tín hiệu phù hợp thì nó sẽ tạo liên kết đến dịch đấy.

**CHƯƠNG 4: KẾT LUẬN**

Kiến thức thu được, kinh nghiệm, hướng phát triển?

**TÀI LIỆU THAM KHẢO**

[1] Giovanni Caire , 2009. Jade programing for beginers.pdf. TILAB, formerly CSELT

[2] Loạt bài hướng dẫn lập trình hướng agent - <https://www.youtube.com/watch?v=VHt9A9Jn1Jk&list=PL6_53ptjaqs4zs-xSK7SwabqC-CjubpIV>

[3]