

JADE

Java Agent DEvelopment Framework

ĐI H C CÔNG NGH THÔNG TIN – ĐHQGTPHCM
LP CNTN02 – KHOA KHOA H C MÁY TÍNH

Gi ng viên h ng d n: ThS. Nguyễn Tr n Minh Khuê

Sinh viên th c hi n:

1. Nguyễn H u Nh t - 07520437
2. Nguyễn Th Thu Trang - 07520594

Mục Lục

1. Giới thiệu	3
1.1. Middleware	3
1.2. Kỹ thuật trong Java.....	4
2. Đặc điểm kỹ thuật FIPA	5
3. Tìm hiểu sơ lược về JADE	9
Kiến trúc của agent platform	10
4. JADE runtime system.....	12
4.1. Sự phân bổ Agent platform.....	12
4.2. Hệ thống con hỗ trợ phân phối thông điệp.....	13
4.3. Quản lý và lưu trữ địa chỉ	14
4.4. Tính di động.....	14
4.5 Người dùng định nghĩa các ontology và nội dung ngôn ngữ	15
4.6. Công cụ cho quản lý và phân định nền tảng	16
6. Tổng quan phương pháp phân tích và thiết kế mô hình đa tác tử sử dụng JADE	16
7. Tổng kết	19
Tài liệu tham khảo.....	20

1. Giới thiệu

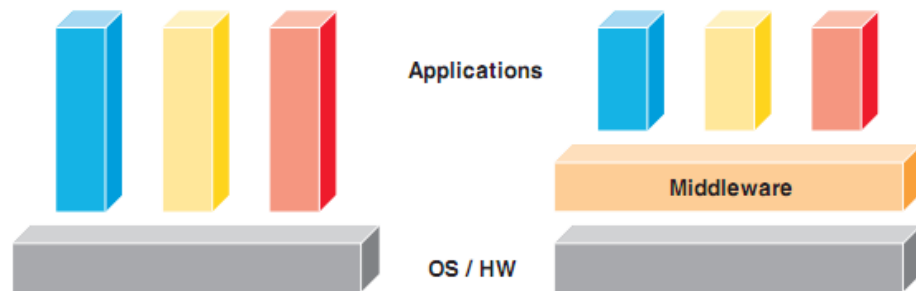
Sự phát triển của nguồn tài nguyên thông tin mạng yêu cầu hệ thống thông tin phải được sắp xếp lại trên hệ thống mạng và tương tác được với các hệ thống khác. Các hệ thống như vậy không dễ dàng được nhận ra với những kỹ thuật phần mềm truyền thống vì sự hạn chế của những kỹ thuật này trong việc sao chép rải rác và có khả năng tương tác. Kỹ thuật dựa trên cơ sở tác tử đường như là một câu trả lời tốt để dễ dàng nhận ra những hệ thống như thế này vì chúng được sáng tạo ra để thực hiện sao chép với những dữ liệu phân tán và có khả năng tương tác.

Kỹ thuật dựa trên tác tử hiện vẫn chưa hoàn thiện và thực sự chỉ mới có một vài hệ thống đáp ứng được nhu cầu trên. Kỹ thuật dựa trên tác tử chưa được sử dụng triệt để hết khả năng của chúng, và sẽ không trở nên phổ biến cho tới khi có một chuẩn cung cấp cho tác tử có khả năng tương tác và được sử dụng bởi những nhà phát triển tác tử trong môi trường tương ứng. Một vài nhà nghiên cứu đang làm việc để đưa ra phương pháp chuẩn hóa cho kỹ thuật tác tử và thực hiện trong môi trường phát triển để xây dựng hệ thống tác tử. Những môi trường phát triển như thế này cung cấp một vài mô hình tác tử và các công cụ được định nghĩa trước để tiện cho sự phát triển của hệ thống. Hơn nữa, một vài trong số chúng cố gắng cho phép tương tác với những hệ thống tác tử khác thông qua việc sử dụng một ngôn ngữ truyền đạt tác tử phổ biến là KQML[3]. Tuy nhiên việc sử dụng phổ biến một ngôn ngữ truyền đạt chung không đủ để cung cấp dễ dàng khả năng tương tác giữa những hệ thống tác tử khác nhau. Việc chuẩn hóa của FIPA là dựa trên phương diện cho phép tương tác dễ dàng giữa các hệ thống tác tử, vì FIPA, vượt ra ngoài ngôn ngữ truyền đạt tác tử, cũng là quy định cụ thể các tác tử chính cần thiết cho việc quản lý một hệ thống tác tử, các ontology cần thiết cho sự tương tác giữa các hệ thống, và nó cũng xác định mức độ vận chuyển của giao thức.

1.1. Middleware

Thuật ngữ *Middleware* có nghĩa là để mô tả tất cả những thư viện cấp cao cho phép phát triển ứng dụng bởi việc cung cấp những dịch vụ chung hữu ích không chỉ cho một ứng dụng đơn lẻ mà là cho một loạt các ứng dụng dễ dàng và hiệu quả hơn, ví dụ thông tin liên lạc, truy cập dữ liệu, mã hóa, kiểm soát nguồn tài nguyên. Những dịch vụ này cùng được cung cấp bởi hệ thống tích hợp, nhưng ý tưởng đằng sau middleware là cung cấp tốt hơn, hệ điều hành độc lập trên sở tích hợp các API có nguồn gốc thành các khối đơn giản để sử dụng lại. Mặt khác, việc thực hiện các dịch vụ này thường đòi hỏi lượng thời gian đáng kể, nếu không giống nhau, trong phương diện hướng tới sự phát triển của ứng dụng logic. Khả năng có thể dùng lại trên nhiều lĩnh vực ứng dụng cho thấy “Horizontal” tiếp cận trái ngược với cách tiếp cận của “Vertical”, nơi một giải pháp Adhoc cho một ứng dụng cụ thể cần được cung cấp. Phương pháp tiếp cận dựa trên Middleware cho phép để giảm footprint và thời gian phát triển của các ứng dụng.

Figure 4
Role of the middleware: "vertical" approach (left) vs. "horizontal" approach (right)



1.2. Kỹ thuật trong Java

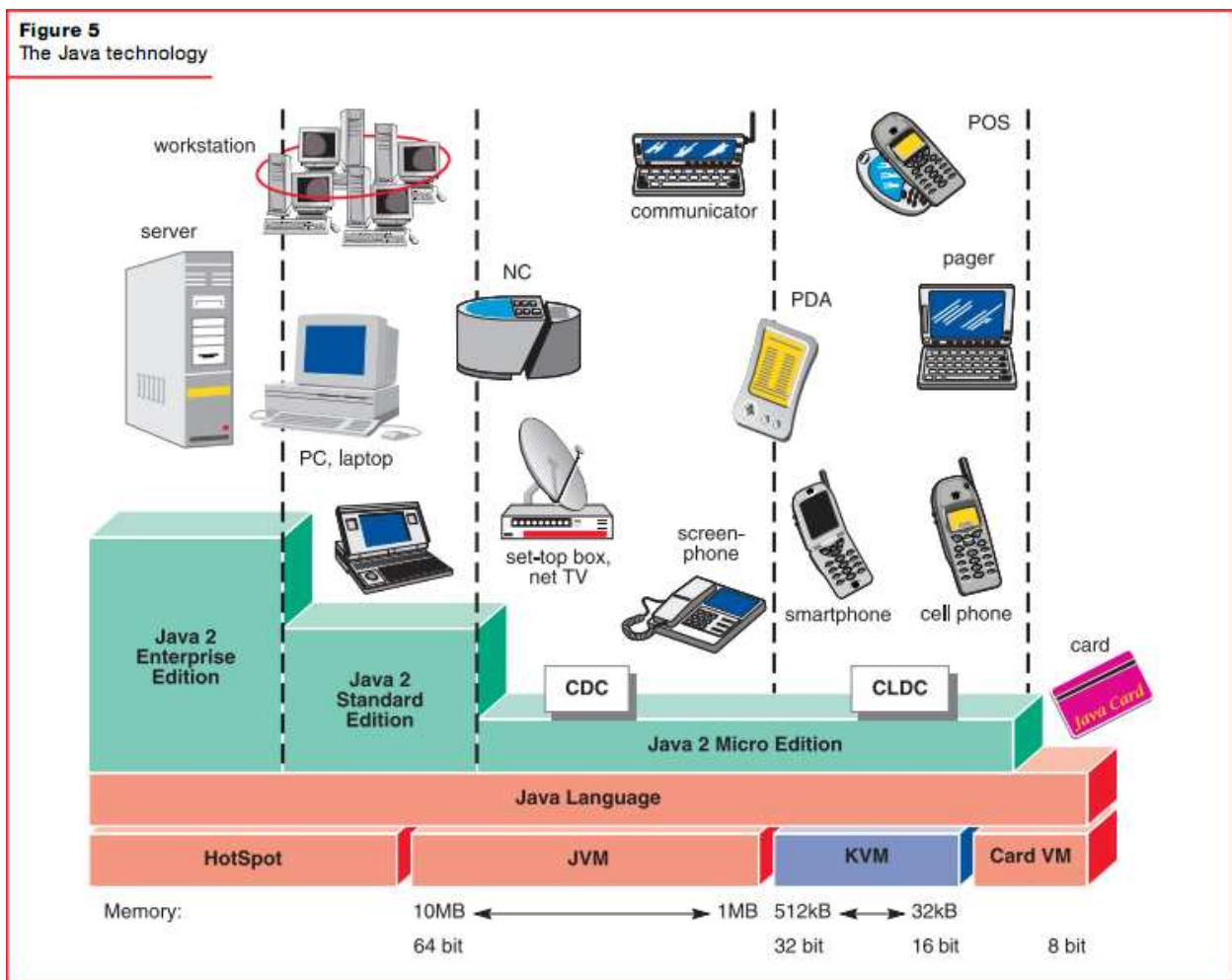
Để hiểu hơn về một số tính năng của Jade và mối quan hệ của nó với thế giới Java thì cần có cái nhìn tổng quát về kỹ thuật Java.

Công nghệ Java được cấu trúc thành 4 phiên bản (được đặt tên bởi Sun) dựa theo mục tiêu và các thiết bị hỗ trợ chức năng:

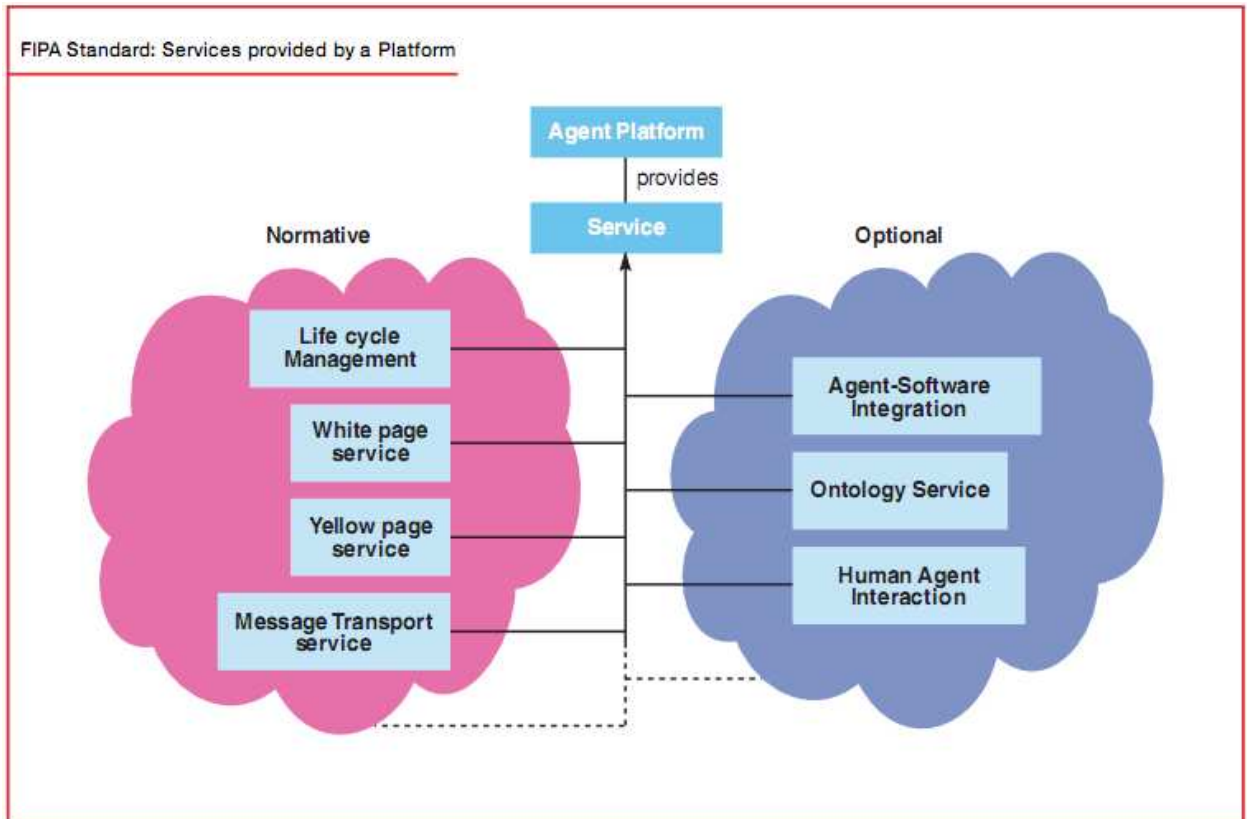
- Các ứng dụng dựa trên máy chủ (J2EE).
- Kiểu ứng dụng máy tính để bàn (J2SE).
- Các ứng dụng xách tay và các thiết bị điện thoại di động (J2ME).
- Dịch vụ SIM / thẻ thông minh (Java Card).

Jade đã được trang bị đầy đủ trong các ngôn ngữ Java, và nó có thể được thực hiện liên tục trên tất cả các loại máy ảo Java ngoại trừ Java Card.

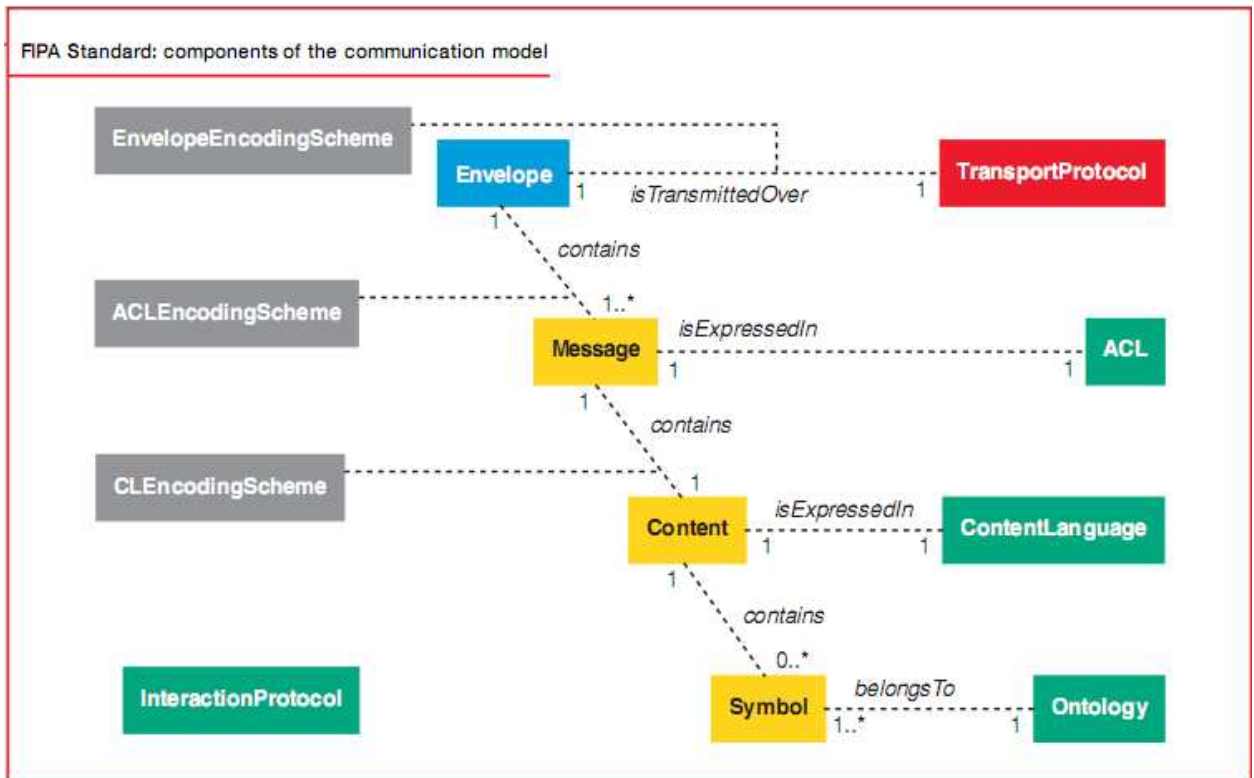
Figure 5
The Java technology



2. Đặc điểm kỹ thuật FIPA



Foundation for Intelligent Physical Agents (FIPA) [5] là một hiệp hội phi lợi nhuận quốc tế của các công ty và các tổ chức chia sẻ các kết quả đạt được để sản xuất các kỹ thuật chung của công nghệ đa tác tử. FIPA được dự kiến không chỉ là kỹ thuật cho một ứng dụng mà còn là kỹ thuật chung cho các lĩnh vực ứng dụng khác, không chỉ là một công nghệ độc lập mà còn là một tập hợp các công nghệ cơ bản có thể được tích hợp bởi các nhà phát triển sử dụng trong một hệ thống phức tạp với một mức độ thích hợp cao.



FIPA dựa trên hai giả định chính.

- Thời gian để đạt được sự đồng thuận và để hoàn tất chuẩn không nên dài, và chủ yếu là, nó không nên hoạt động như một brake trong tiến trình hơn là tạo một khả năng có thể.
- Hành vi bên ngoài của các thành phần hệ thống nên được quy định cụ thể, để lại chi tiết thực hiện và kiến trúc nội bộ để phát triển tác tử.

Trong thực tế, kiến trúc nội tại của Jade là sở hữu độc quyền ngay cả khi nó phù hợp với các giao diện được chỉ định bởi FIPA.

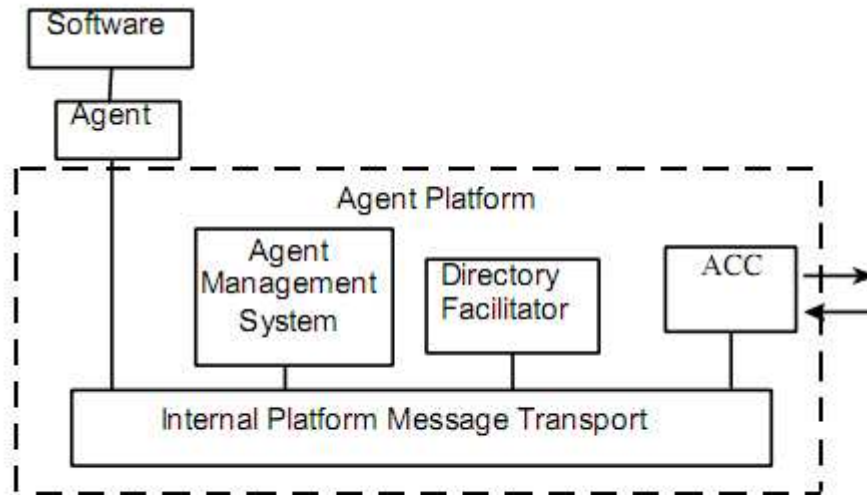
Các tài liệu đầu ra đầu tiên của FIPA, được gọi là kỹ thuật FIPA97, xác định các quy tắc quy phạm cho phép một xã hội các tác tử có khả năng tương tác có thể hoạt động và được quản lý. Trước hết tất cả chúng mô tả các mô hình tham chiếu của một nền tảng tác tử (Agent Platform). Về cơ bản, nó xác định vai trò của một số tác tử chính cần thiết cho việc quản lý Platform này, và quy định cụ thể tác tử quản lý nội dung ngôn ngữ và ontology. Ba vai trò quan trọng bắt buộc đã được xác định vào một *Agent Platform*. Hệ thống quản lý tác tử - Agent Management System (AMS), là tác nhân tác động điều khiển giám sát qua tiếp cận và sử dụng Platform này, nó có trách nhiệm xác thực các tác tử thường trú và kiểm soát đăng ký. Agent Communication Channel (ACC) là tác tử cung cấp đường dẫn cho các liên hệ cơ bản giữa các tác tử bên trong và bên ngoài Platform; nó là phương pháp truyền thông mặc định mà cung cấp một

dịch vụ có thứ tự và đáng tin cậy; nó cũng phải hỗ trợ cho khả năng tương tác IIOP giữa các Agent Platform khác nhau. Facilitator Directory (DF) là tác tử cung cấp một dịch vụ *Yellow Page* cho các Agent Platform. Chú ý rằng không có sự giới hạn trong các công nghệ hiện nay sử dụng cho việc thực hiện Platform : e-mail dựa trên platform, CORBA, ứng dụng Java multi-thread, ... tất cả có thể được FIPA triển khai.

Tất nhiên, tiêu chuẩn quy định cụ thể cũng là Agent Communication Language (ACL). Tác tử giao tiếp dựa trên truyền qua tin nhắn (message transport), nơi các tác tử giao tiếp bằng cách lập và gửi tin nhắn cá nhân với nhau. Các ACL FIPA quy định cụ thể một ngôn ngữ tin nhắn chuẩn bằng cách đặt ra tiêu chuẩn mã hóa, ngữ nghĩa và pragmatics của tin nhắn. Tiêu chuẩn này không đặt ra một cơ chế cụ thể cho việc vận chuyển nội bộ của tin nhắn. Thay vào đó, kể từ khi các tác tử khác nhau có thể chạy trên những platforms khác nhau và sử dụng công nghệ mạng khác nhau, FIPA quy định rằng các thông điệp vận chuyển giữa các platform nên được mã hóa trong một hình thức văn bản. Giả định rằng tác tử có một số phương tiện truyền văn bản dạng này. Cú pháp của ACL là rất gần với ngôn ngữ sử dụng rộng rãi KQML truyền thông. Tuy nhiên, mặc dù cú pháp tương tự, nhưng vẫn có những khác biệt cơ bản giữa KQML và ACL, hiển nhiên nhất là sự tồn tại của một ngữ nghĩa chính thức cho ACL mà nên loại bỏ bất cứ sự không rõ ràng và nhầm lẫn từ việc sử dụng ngôn ngữ.

Tiêu chuẩn hỗ trợ các hình thức phổ biến của các cuộc hội thoại giữa các agent thông qua các đặc điểm kỹ thuật của các giao thức tương tác, đó là mô hình của thư trao đổi của hai hoặc nhiều tác tử. Giao thức này bao gồm từ giao thức truy vấn đơn giản, đến các hợp đồng đàm phán giao thức mạng nổi tiếng.

Các phần khác của tiêu chuẩn FIPA xác định các khía cạnh khác, đặc biệt là việc tích hợp phần mềm tác tử (agent-software), tác tử bảo mật di động, dịch vụ ontology, và truyền thông tác tử con người (Human-Agent Communication). Tuy nhiên, chúng chưa được xem xét thực hiện trong Jade [5].



FIPA reference model of an Agent Platform

3. Tìm hiểu sơ lược về JADE

JADE (Java Agent DEvelopment Framework) là một phần mềm framework được xây dựng hoàn toàn bằng ngôn ngữ Java. Nó đơn giản hoá việc triển khai các hệ thống đa tác tử (multi-agent systems) thông qua một *middleware* mà nó đòi hỏi phải tuân theo các chi tiết kỹ thuật FIPA và thông qua một bộ công cụ hỗ trợ trong gỡ lỗi và trong triển khai các giai đoạn. Nền tảng tác tử (agent platform) có thể được phân phối trên nhiều máy (không nhất thiết cần phải cùng một hệ điều hành) và cấu hình có thể được điều khiển thông qua một giao diện từ xa (a remote GUI). Cấu hình có thể được thay đổi ngay cả ở thời gian chạy (run-time) bằng cách tạo ra các agent mới và di chuyển các agent từ một máy đến một máy khác khi cần thiết. Yêu cầu của hệ thống chỉ cần có The Java Run Time version 5.0 trở lên. Các kiến trúc truyền thông cung cấp truyền tin (messaging) linh hoạt và hiệu quả, JADE tạo ra và quản lý một hàng đợi các tin nhắn ACL đến từng agent. Các agent có thể truy cập vào hàng đợi thông qua sự kết hợp của nhiều phương thức: blocking, polling, time-out và mô hình kết hợp cơ bản (pattern match based). Mô hình đầy đủ của FIPA communication đã được hỗ trợ và các thành phần của nó đã được tích hợp phân biệt rõ ràng và đầy đủ: các giao thức tương tác (interaction protocols), đóng gói (envelope), ACL, các ngôn ngữ thích hợp (content languages), quá trình mã hóa (encoding schemes), các ontology và giao thức vận chuyển (transport protocols). Cơ chế vận chuyển thích nghi với từng tính huống riêng biệt trong quá trình lựa chọn giao thức tốt nhất. Hầu hết các giao thức tương tác được định nghĩa bởi FIPA đã có sẵn và được trang bị sau khi xác định yêu cầu của ứng dụng đối với mỗi giao thức. SL và ontology quản lý tác tử (agent management ontology) đã được bổ sung, cũng như hỗ trợ các ngôn ngữ thích hợp do người dùng định nghĩa và các ontology có thể được trang bị, đăng ký và sử dụng một cách tự động bởi framework.

Mục tiêu của Jade là để đơn giản hóa sự phát triển trong khi vẫn đảm bảo tuân thủ tiêu chuẩn thông qua một hệ thống dịch vụ và tác tử. Để đạt được mục tiêu trên, Jade cung cấp danh sách của các tính năng cho việc lập trình tác tử sau đây:

- *FIPA-compliant Agent Platform*, trong đó bao gồm các AMS (Hệ thống quản lý tác tử - Agent Management System), các DF (Directory Facilitator), và ACC (kênh truyền thông tác tử - Agent Communication Channel). Tất cả ba tác tử trên sẽ được tự động kích hoạt khi agent platform được kích hoạt.

- *Agent Platform phân phối*: agent platform có thể được chia một số trên máy (với điều kiện là không có tường lửa giữa chúng). Chỉ có một ứng dụng Java, và do đó chỉ có một máy ảo Java (Java Virtual Machine) được thực hiện trên mỗi máy chủ. Tác tử được thực hiện như một Java thread và Java event được sử dụng cho việc truyền thông nhẹ nhàng và hiệu quả giữa các tác tử trên cùng một máy chủ. Những nhiệm vụ song song vẫn còn có thể được thực hiện bởi một tác tử, và lịch trình Jade các nhiệm vụ này trong một cách hiệu quả hơn (và thậm chí đơn giản hơn cho các lập trình viên có tay nghề) so với Java Virtual Machine làm đối với các thread.

- Một số *FIPA-compliant DF* (Directory Facilitator) có thể được bắt đầu vào thời điểm chạy để thực hiện đa miền ứng dụng.

- Lập trình giao diện để đơn giản hóa đăng ký dịch vụ tác tử với một, hoặc nhiều domains (ví dụ như DF)

- Cơ chế vận chuyển và giao tiếp để gửi/nhận tin nhắn giữa các tác tử với nhau
- Tuân thủ giao thức IIOP-FIPA97 để kết nối các Agent Platforms khác nhau
- Phương thức trao đổi đơn giản của ACL message bên trong các Agent Platform giống nhau, như thông điệp được chuyển mã hóa là đối tượng Java, hơn là chuỗi, để tránh thủ tục marshalling và unmarshalling. Khi người gửi hoặc nhận không thuộc cùng một platform, thông điệp sẽ được tự động chuyển sang các định dạng chuỗi FIPA

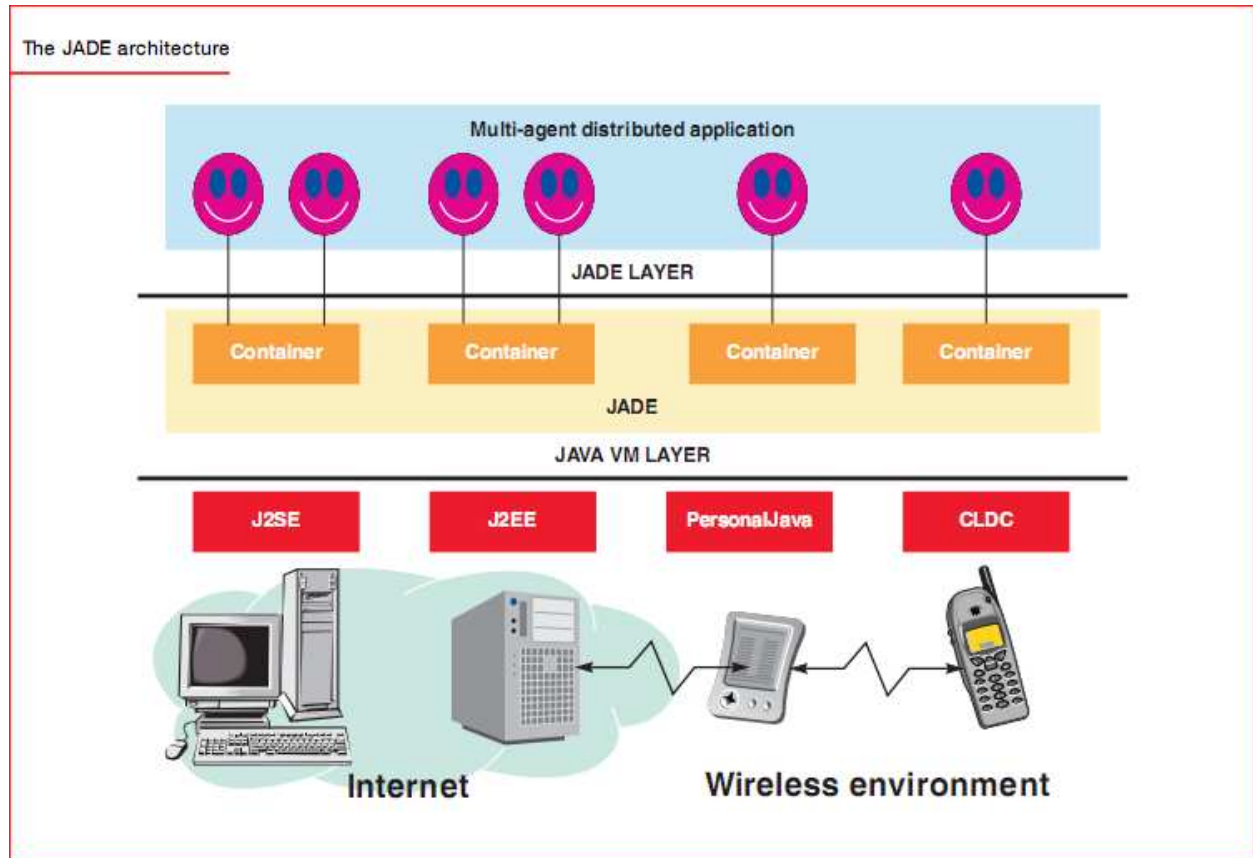
- Thư viện các giao thức tương tác FIPA sẵn sàng để được sử dụng
- Đăng ký tự động của tác tử với AMS
- Tên dịch vụ tương thích FIPA: lúc khởi động các tác tử tồn tại luôn GUID (Globally Unique Identifier) của chúng từ platform.

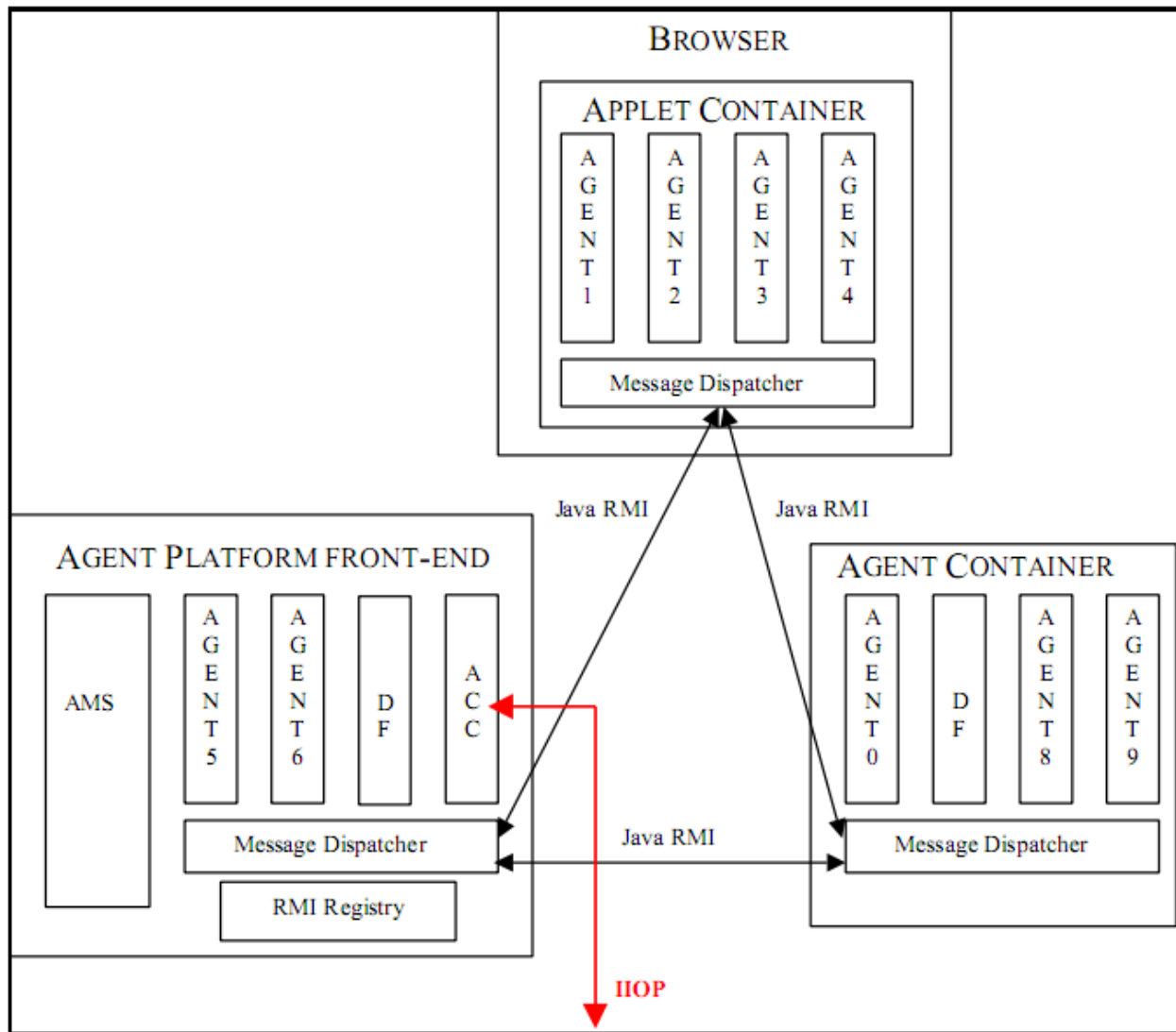
- Đồ họa giao diện người dùng để quản lý một số tác tử và agent platforms từ cùng một tác tử. Tình hoạt động của platform có thể được theo dõi và khóa

- Giao diện đồ họa cho người dùng để quản lý nhiều tác tử và các platforms tác tử từ cùng một tác tử.

Kiến trúc của Agent Platform

JADE Agent Platform dựa theo kỹ thuật FIPA97 và bao gồm tất cả các mandatory agents quản lý platform, đó là ACC, các AMS, và DF. Mọi liên lạc của các tác tử được thực hiện thông qua tin nhắn (message), nơi FIPA ACL là ngôn ngữ để trình bày tin nhắn.





Software Architecture of one JADE Agent Platform

Các kiến trúc phần mềm dựa trên sự cùng tồn tại của một số máy ảo Java (Java Virtual Machines) và giao tiếp dựa trên Java RMI (Remote Method Invocation) giữa các máy ảo khác nhau và sự kiện báo hiệu bên trong một máy ảo duy nhất. Mỗi máy ảo là một container cơ bản của tác tử, cung cấp một môi trường thời gian chạy hoàn chỉnh cho tác tử thi hành và cho phép thực hiện một số tác tử để thực hiện đồng thời trên cùng một host. Về nguyên tắc, kiến trúc cũng cho phép nhiều máy ảo được thực hiện trên cùng một máy chủ.

Mỗi tác tử container là một môi trường thực thi đa luồng gồm một thread cho tất cả các tác tử cộng với hệ thống thread sinh ra bởi hệ thống RMI runtime cho việc gửi tin nhắn. Một container đặc biệt đóng vai trò front-end, quản lý hoạt động tác tử và trình bày cho toàn bộ platform tới thế giới bên ngoài. Một Agent Platform (AP) hoàn thành là sau đó bao gồm một vài tác tử containers. Sự phân phối container trên một mạng máy tính được cho phép, với điều kiện

RMI truyền thông giữa các máy của họ được bảo tồn. Một container hạng nhẹ đặc biệt đang được triển khai cho việc thi hành các tác tử trong một trình duyệt Web.

Mỗi Agent Container là một đối tượng RMI server quản lý cục bộ một tập hợp các tác tử. Nó điều khiển vòng đời của tác tử bằng cách tạo ra, tạm dừng, khôi phục và hủy chúng. Bên cạnh đó, những giao dịch với tất cả các khía cạnh giao tiếp bằng các thông điệp ACL, định tuyến theo lĩnh vực đích và đưa chúng vào hàng đợi tin nhắn thông điệp tác tử riêng; đối với thông điệp gửi đi, thay vào đó, các tác tử container duy trì đủ thông tin để tìm kiếm vị trí tác tử nhận và chọn một phương pháp phù hợp để chuyển tiếp thông điệp ACL.

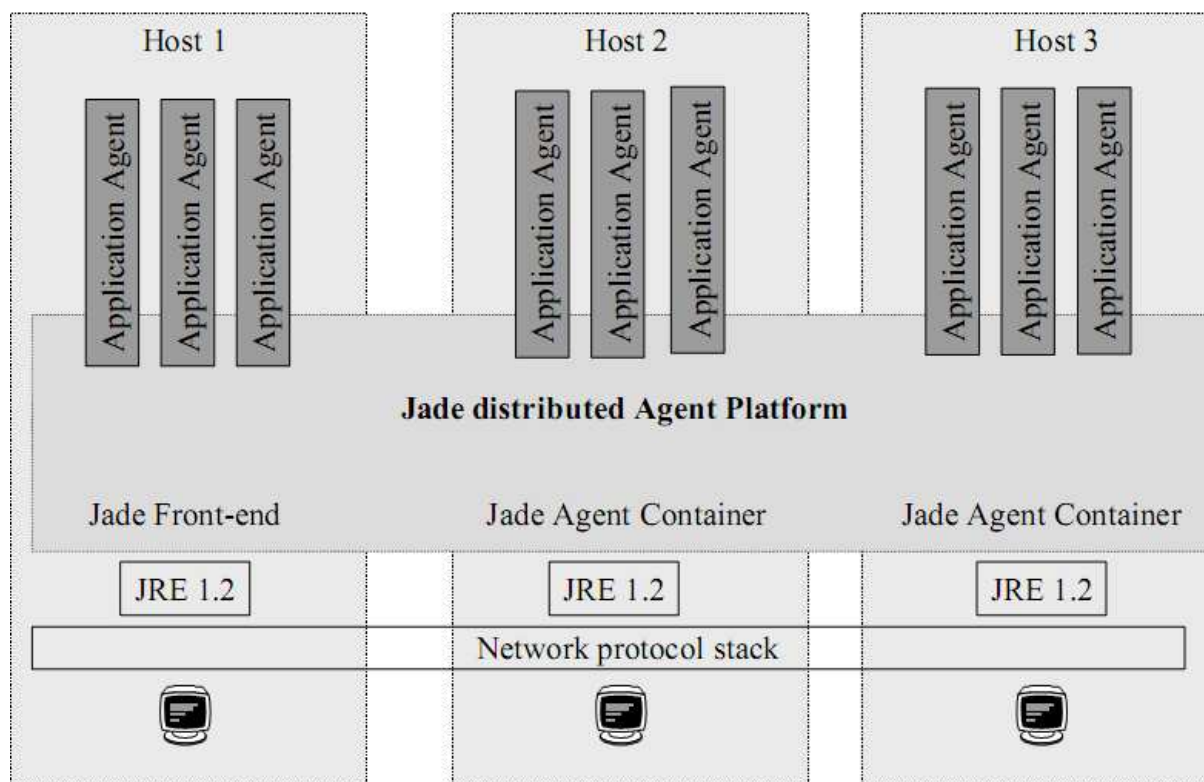
Các agent platform cung cấp một giao diện đồ họa người dùng (Graphical User Interface - GUI) cho việc quản lý từ xa, kiểm tra và kiểm soát tình trạng của các tác tử (ví dụ như dừng và khởi động lại các tác tử). GUI cũng cho phép đề tạo ra và bắt đầu việc thực thi các tác tử trên một máy chủ từ xa, với điều kiện là có một tác tử container đang chạy. GUI được thực hiện như một tác tử, gọi là RMA (Remote Monitoring Agent). Tất cả các thông tin liên lạc giữa các tác tử và GUI, tất cả các thông tin liên lạc giữa GUI và AMS được thực hiện thông qua ACL thông qua một phần mở rộng Adhoc của ontology Fipa-agent-management. Phần mở rộng này sẽ được đưa ra FIPA xem xét để chuẩn hóa. Nếu nó được chấp nhận, các tác tử RMA có thể cùng sử dụng để kiểm soát những Agent Platforms khác, bao gồm các platform không phải Jade.

4. JADE run-time system

4.1. Sự phân bổ Agent platform:

Jade tuân theo các chi tiết kỹ thuật của FIPA97 và bao gồm tất cả các hệ thống tác tử quản lý nền tảng, đó là ACC, AMS, và DF mặc định. Tất cả các giao tiếp giữa tác tử được thực hiện thông qua các tin nhắn/thông điệp, FIPA ACL là ngôn ngữ được sử dụng để trình bày cho các thông điệp.

Khi xuất hiện như một thực thể duy nhất với thế giới bên ngoài, một nền tảng tác tử JADE (JADE agent platform) là một hệ thống phân phối của chính nó, có thể được phân chia thành nhiều chủ thể (host) với một trong số chúng hoạt động cho nền tảng quá trình liên lạc IIOP. Một hệ thống JADE được tạo nên từ một hoặc nhiều *agent container*, mỗi *agent container* hoạt động trong một máy ảo Java và giao tiếp bằng cách sử dụng Java RMI. IIOP được sử dụng để chuyển tiếp đi các thông điệp đến các nền tảng tác tử bên ngoài. *Front End container* cũng là một máy chủ IIOP. Hình sau cho thấy kiến trúc của một JADE Agent Platform.



Hình : Kiến trúc phần mềm của JADE Agent Platform

4.2. Hệ thống con hỗ trợ phân phối thông điệp:

Mô hình giao tiếp tác tử FIPA là dạng peer-peer-to thông qua các ngữ cảnh đa thông điệp được cung cấp bởi các giao thức tương tác và định danh hội thoại (conversation identifiers). Mặt khác, JADE sử dụng phương pháp vận chuyển như RMI, CORBA và việc gửi đi sự kiện với đặc trưng liên kết với các hệ thống phản ứng. Có một vài lỗ hổng bắt cầu cho bản đồ đánh địa chỉ mô hình FIPA tin truyền một cách rõ ràng trong yêu cầu / đáp ứng mô hình thông tin liên lạc của các đối tượng được phân phối. Cho thấy lý do tại sao các tác tử thông thường của JADE không được phân phối còn các *agent container* thì được.

Một phần mềm phù hợp với mô hình tác tử của FIPA, có một *Globally-unique identifier* (GUID), cái mà có thể được sử dụng bởi tất cả các tác tử khác để đánh địa chỉ với thông điệp ACL; một tác tử sẽ đặt GUID của nó vào một *sender slot* của thông điệp ACL. JADE phải chỉ ra vị trí *receiver* bằng cách xem xét *receiver slot*. Một FIPA97 GUID giống với một địa chỉ email có dạng: <tên tác tử> @ <địa chỉ nền tảng>, giúp khá dễ dàng để khôi phục lại tên tác tử và địa chỉ các nền tảng từ nó. Khi một thông điệp ACL được gửi đến một phần mềm tác tử, ba tùy chọn được đưa ra:

- *Receiver* trên cùng *container* của cùng nền tảng: các Java event được sử dụng, *ACLMessage* được nhân bản.
- *Receiver* trên một *container* khác nhau của cùng một nền tảng: Java RMI được sử dụng, thông điệp được phát ở nơi *sender*, một phương thức từ xa được gọi và thông điệp không phát nơi *receiver*.
- *Receiver* trên một nền tảng khác nhau: IIOP được sử dụng, *ACLMessage* được chuyển đổi sang một *String* và sắp xếp theo thứ tự ở nơi *sender*, một CORBA từ xa được thực hiện và không sắp xếp theo thứ tự theo sau quá trình phân tích cú pháp ACL xảy ra ở *receiver*.

4.3. Quản lý và lưu trữ địa chỉ:

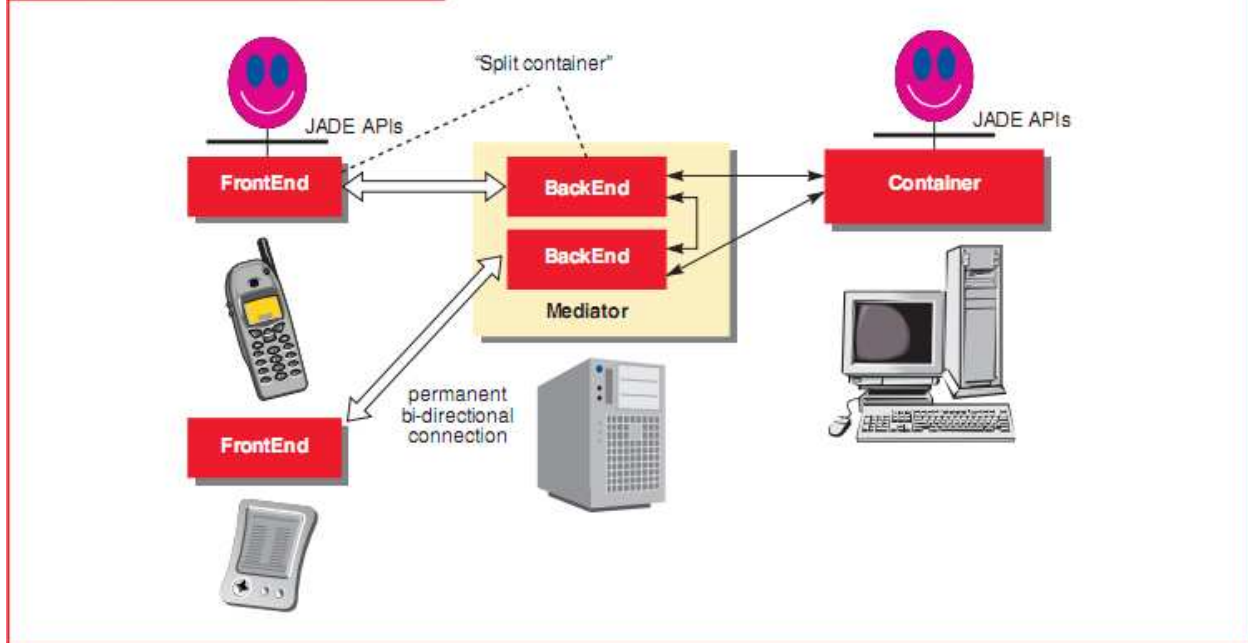
JADE sẽ chọn cơ chế thuận tiện nhất của 3 cơ chế phân phối ở trên theo vị trí tác tử. Về cơ bản, mỗi *container* có một bảng các tác tử cục bộ của mình, gọi là các *Local-Agent Descriptor Table* (LADT), bên cạnh đó *container* cũng duy trì Global-Agent Descriptor Table (GADT), lập bản đồ (mapping) mỗi tác tử, tham chiếu đối tượng RMI *container* của nó. JADE sử dụng một kỹ thuật lưu địa chỉ để tránh các truy vấn liên tục cho thông tin địa chỉ. Bên cạnh đó là tính hiệu quả, để hỗ trợ tác tử di động (agent mobility) nơi mà địa chỉ tác tử có thể thay đổi theo thời gian.

4.4. Tính di động:

Phiên bản mới của JADE có thêm sự hỗ trợ cho tính di động tác tử. Khai thác Java Serialization API và dynamic class loading, có thể di chuyển hoặc sao chép một JADE agent trong cùng một nền tảng tác tử của JADE. Chi tiết kỹ thuật FIPA để được hỗ trợ di động vẫn còn chưa đầy đủ, chỉ dịch vụ di động sẽ không giúp tiêu chuẩn hóa trong khả năng tương tác. Một số đề nghị tổng quát hơn sẽ được trình FIPA, trải qua công khai thảo luận và đánh giá. Sau đó, một công cụ hỗ trợ có hiệu quả và tương thích có thể được xây dựng.

JADE run-time có thể được thực thi trên một lớp rộng các thiết bị khác nhau, từ máy chủ tới điện thoại di động.

JADE architecture in the wireless environment



Để đúng địa chỉ bộ nhớ và hạn chế sức mạnh xử lý của thiết bị di động và các đặc tính của mạng không dây (GPRS nói riêng) về băng thông, độ trễ, kết nối liên tục và biến đổi địa chỉ IP, và đồng thời để có hiệu quả khi thực hiện trên máy mạng cố định, Jade có thể được cấu hình để thích ứng với các đặc tính của môi trường triển khai. Kiến trúc Jade hoàn toàn kiểu module và, bằng cách kích hoạt các module nhất định thay cho những thứ khác, có thể để đáp ứng các yêu cầu khác nhau về kết nối, bộ nhớ và sức mạnh xử lý.

Một module gọi là LEAP cho phép tối ưu hóa tất cả các cơ chế giao tiếp khi xử lý các thiết bị với nguồn lực hạn chế và kết nối thông qua mạng không dây. Bằng việc kích hoạt module này, một container Jade là “split” thực sự đang chạy trên các thiết bị đầu cuối di động, và lại chạy trong mạng cố định.

4.5. Người dùng định nghĩa các ontology và nội dung ngôn ngữ:

Theo tiêu chuẩn FIPA, để đạt mức tương hợp các tác tử yêu cầu các tác tử khác nhau phải chia sẻ nhiều hơn một giao thức đơn giản. Trong khi FIPA ủy quyền cho FIPA ACL, nó cho phép ứng dụng phụ thuộc vào nội dung ngôn ngữ và ontology. Các chi tiết kỹ thuật FIPA chứa một thư viện nội dung ngôn ngữ, trong khi đó sự đa dạng của các ontology được định nghĩa và được sử dụng trong các phần khác nhau của tiêu chuẩn FIPA.

Phiên bản cuối cùng của JADE cho phép lập trình ứng dụng tạo ra nội dung ngôn ngữ và ontology của riêng mình. Mỗi JADE agent có bao gồm khả năng lực hiểu biết đến ngôn ngữ và

ontology được liệt kê; người dùng định nghĩa code phải có khả năng dịch qua lại giữa các định dạng String (theo cú pháp nội dung ngôn ngữ) và một đại diện cơ sở.

Nếu một ontology được người dùng định nghĩa, các ứng dụng có thể đăng ký một lớp Java phù hợp đóng vai trò bản thể và JADE có thể chuyển đổi đến và đi từ khung và người dùng định nghĩa đối tượng Java. Làm theo cách này, các lập trình ứng dụng có thể định nghĩa khái niệm tên miền cụ thể của họ như là lớp Java quen thuộc, trong khi vẫn có khả năng quá trình đó ở cấp tác tử.

4.6. Công cụ cho quản lý và phân định nền tảng:

Ngoài một thư viện runtime, JADE còn cung cấp một số công cụ để quản lý các nền tảng tác tử, theo dõi và gỡ lỗi cho xã hội tác tử; tất cả những công cụ này được thực hiện như FIPA agent, chỉ cần dựa vào JADE AMS. Việc quản lý giao diện điều khiển chung cho một nền tảng JADE được gọi là tác tử RMA (Remote Monitoring Agent). RMA đạt được các thông tin về nền tảng và thực hiện các lệnh của giao diện đồ họa (GUI) để sửa đổi trạng thái của nền tảng (tạo ra các tác tử, đóng *container*, ...v.v...) thông qua AMS. Các tác tử Directory Facilitator cũng có một giao diện, mà nó có thể được quản lý, cấu hình các tác tử.

Người dùng JADE có thể gỡ lỗi tác tử của họ với các Dummy Agent và the Sniffer Agent. Dummy Agent là một công cụ đơn giản để kiểm tra thông điệp trao đổi giữa các tác tử, tạo điều kiện xác nhận mô hình trao đổi thông điệp và tương tác thử nghiệm của các tác tử.

Sniffer Agent cho phép theo dõi các trao đổi thông điệp trong một nền tảng tác tử JADE: mỗi thông điệp trực tiếp tới hoặc đến từ một tác tử đã chọn hay nhóm được theo dõi và hiển thị trong cửa sổ sniffer, sử dụng một ký hiệu tương tự như *UML Sequence Diagrams*.

5. Tổng quan phương pháp phân tích và thiết kế mô hình đa tác tử sử dụng JADE

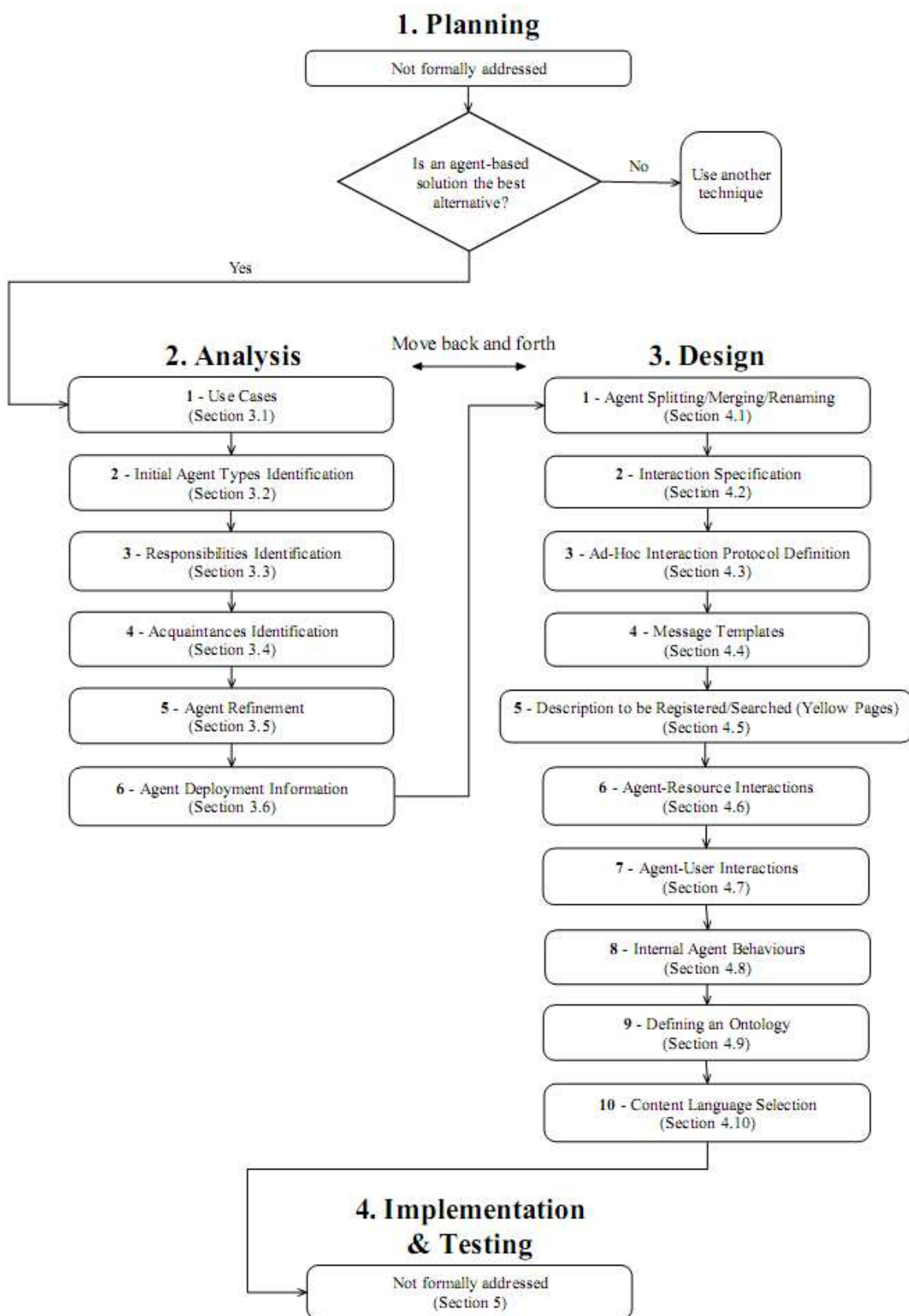
Một phương pháp hỗ trợ như một hướng dẫn các nhà thiết kế khi phát triển một hệ thống đa tác tử. Nói chung, một phương pháp phát triển phần mềm gồm có:

- Một quy trình, nghĩa là một chuỗi các giai đoạn và các bước hướng dẫn các nhà phát triển xây dựng hệ thống.
- Một tập hợp các quy tắc heuristic có hỗ trợ nhà phát triển trong việc đưa ra sự lựa chọn có liên quan.
- Một số sơ đồ, lược đồ hoặc các văn bản mô phỏng trong đồ họa hoặc hình thức văn bản của một hoặc nhiều mô hình hệ thống.
- Một ký hiệu phù hợp sẽ được sử dụng trong các tác vụ.
- Một tập hợp các mô hình có thể được áp dụng để giải quyết các tình huống thường gặp.

- Một hoặc nhiều công cụ: tự động hoá, càng nhiều càng tốt, các giai đoạn và các bước quy định cụ thể trong quá trình này; thống nhất giữa các mô hình sản xuất; nêu bật các vấn đề phát sinh từ sự lựa chọn thiết kế không đúng, khi có thể; tạo code và tài liệu, ...

Trọng tâm hiện nay của phương pháp được đề xuất là về quy trình và các tác vụ được thực hiện. Một bản thảo các ký hiệu cũng được giới thiệu sử dụng trong xây dựng các tác vụ, một số quy tắc heuristic và mẫu thiết kế được trình bày. Quá trình được mô tả bao gồm giai đoạn phân tích và giai đoạn thiết kế (được mô tả như hình bên dưới).

Giai đoạn phân tích tổng quan một cách tự nhiên và độc lập với nền tảng. Ngược lại, các giai đoạn thiết kế cụ thể giả định là nền tảng JADE thực hiện và tập trung trực tiếp trên các lớp và khái niệm được cung cấp bởi JADE. Có thể thấy rằng không có ranh giới nghiêm ngặt giữa giai đoạn phân tích và thiết kế. Hơn nữa, phương pháp này là của một tính chất lặp đi lặp lại, do đó cho phép các nhà thiết kế di chuyển qua lại giữa các giai đoạn phân tích và các giai đoạn thiết kế ứng với từng bước trong đó.



Hình : tổng quan về phương pháp phân tích và thiết kế hệ thống multi-agent

6. Tổng kết

JADE - một phần mềm framework tạo thuận lợi cho sự phát triển của các ứng dụng tác tử và mô tả kiến trúc phần mềm của nó và nó cung cấp các hàm chức năng cũng như liên kết với một số công nghệ, đặc biệt là dịch vụ web và các công cụ suy luận và mô hình kiến trúc. Có thể có lý do cho sự phát triển của JADE, đặc biệt là tính hữu dụng cũng cho các dự án khác trong tương lai. Jade là một phần mềm công cụ rất phát triển, rất nhiều người đang làm việc trên nó, duy trì liên tục hoạt động của nó, cải tiến và mở rộng, trong công nghiệp và thương mại các ứng dụng được xây dựng và tồn tại dưới nhiều khác nhau dựa trên sự hỗ trợ của JADE framework. Xu hướng trong tương lai là tập trung vào một nền tảng cấu hình và các mô-đun điều khiển như trong kiến trúc của hệ thống đa tác tử cho việc phát triển ứng dụng, ứng như sự phát triển của một quy trình làm việc dựa trên cơ chế phân phối các engine. Engine mới này sẽ cho phép agent thực thi các ứng dụng logic mô tả thông qua việc phân luồng công việc, và tác động lại các agent khác của quy trình công việc theo tiêu chuẩn được đánh giá ở thời gian chạy. Các tiêu chuẩn đó có thể dựa trên ví dụ về khối lượng công việc của agent hiện tại hoặc đặc điểm cụ thể của agent. Đây là động cơ cho việc xử lý cùng với một dịch vụ tích hợp tạo nên môi trường hỗ trợ các quá trình công việc trên một giao diện đồ họa thân thiện với người dùng, đây sẽ là nòng cốt tạo ra giá trị của phiên bản mới của JADE. Ta có thể tranh luận: nếu JADE là một platform có hỗ trợ mạnh, tại sao không phải là nó làm agent theo định hướng lập trình cho một tiêu chuẩn kỹ thuật trong ngành công nghiệp phần mềm? Jade là một middleware cho phép thực hiện nhanh chóng và đáng tin cậy của hệ thống đa tác tử và có thể được tích hợp với công cụ trí tuệ nhân tạo (AI), tuy nhiên, lợi ích của AI và các agent hợp nhất hơn trong cộng đồng khoa học hơn là trong ngành công nghiệp phần mềm. Jade có thể được coi là agent platform phổ biến nhất sẵn có hiện nay.

Tài liệu tham khảo

1. JADE - A FIPA – compliant agent framework , Fabio Bellifemine bellifemine@cse.it , Agostino Poggi poggi@ce.unipr.it, Giovanni Rimassa rimassa@ce.unipr.it .
2. JADE White Paper - F. Bellifemine, G. Caire, A. Poggi, G. Rimassa <http://exp.telecomitalia.com> .
3. JADE: A software framework for developing multi-agent applications - Fabio Bellifemine, Giovanni Caire, Agostino Poggi, Giovanni Rimassa - www.sciencedirect.com .
4. <http://jade.tilab.com/doc/index.html>
5. Developing Multi-agent Systems with JADE - Fabio Bellifemine, Agostino Poggi, Giovanni Rimassa. - bellifemine@cse.it , poggi@ce.unipr.it , rimassa@ce.unipr.it .