

ÁP DỤNG LÝ THUYẾT KIỂU ĐẶC TẢ HÌNH THỨC HỆ THỐNG ĐA TÁC TỬ ĐỆ QUY

APPLYING THEORY OF TYPES TO FORMAL SPECIFICATION OF RECURSIVE MULTI-AGENT SYSTEMS

HOÀNG THỊ THANH HÀ
Trường Đại học Kinh tế, Đại học Đà Nẵng

TÓM TẮT

Một hệ thống phức tạp được xem như tập các hệ thống con. Các hệ thống con này cùng tồn tại và tương tác lẫn nhau. Gần đây, hệ thống đa tác tử - là một dạng của hệ thống phức tạp - rất được phát triển. Trong bài báo này, chúng tôi tập trung vào nghiên cứu hệ thống đa tác tử đệ quy. Đây là mô hình thích hợp để đặc tả các hệ thống phức tạp mang tính đệ quy. Hiện nay, hệ thống đa tác tử đệ quy chỉ được đặc tả bởi những ngôn ngữ phi hình thức. Bài báo này đưa ra đề xuất sử dụng lý thuyết kiểu đặc tả hình thức hệ thống trên.

ABSTRACT

Today, software systems are more and more complex. Such systems are composed of many sub-systems, in which each sub-system exists in interaction with other sub-systems. Recently, multi-agent systems (MAS), one of these kinds of systems, have been studied thoroughly. In this paper, we concentrate on studying the recursive MAS which are well adapted to describe complex systems. Until now, recursive MAS have only been described by non-formal languages. This paper proposes using the theory of mentioned types to specify recursive MAS.

1. Đặt vấn đề

Một hệ thống phức tạp có thể được xem như là một tập hợp các phần tử. Các phần tử này một mặt hỗ trợ cho nhau để hoàn thành nhiệm vụ chung, mặt khác chúng lại phụ thuộc vào các hệ thống con của nó để hoàn thành nhiệm vụ riêng. Để đặc tả các hệ thống trên, sử dụng hệ thống đa tác tử (Multi-Agent System - MAS) có nhiều ưu thế. Các phần tử của hệ thống được xem như các tác tử, tập hợp các phần tử của hệ thống được xem như một MAS. Không chỉ dừng lại là tập hợp các phần tử, các hệ thống phức tạp phải xem xét xuyên qua tất cả các hoạt động của các phần tử cấu thành bằng cách phân rã các phần tử ở mức độ chi tiết hơn. Hay một phần tử ở mức n phân rã thành một tập hợp các phần tử con ở mức $n-1$. Như vậy sự phân rã hệ thống mang tính đệ quy. Vì thế, nhiều nghiên cứu đã đưa khái niệm đệ quy vào trong MAS.

Cho đến nay, MAS đệ quy vẫn chỉ được đặc tả bằng các ngôn ngữ phi hình thức. Mục đích của bài báo này là đề xuất việc sử dụng một ngôn ngữ hình thức để đặc tả MAS đệ quy. Từ đó, chúng ta có thể phân tích, thiết kế, mô phỏng, kiểm thử tính đúng đắn của của hệ thống phức tạp mà hệ thống đó được đặc tả bởi MAS đệ quy.

Phần tiếp theo của bài báo, chúng tôi sẽ sơ lược khái niệm MAS và các tiếp cận đặc tả của chúng. Sau đó mô hình MAS đệ quy sẽ được giới thiệu. Từ đó đề xuất áp dụng lý thuyết kiểu để đặc tả MAS đệ quy sẽ được trình bày. Một ứng dụng cụ thể được trình bày để minh họa cho giải pháp. Cuối cùng bài báo kết thúc bởi kết luận.

2. Hệ thống đa tác tử và các tiếp cận đặc tả hình thức

- Khái niệm tác tử

Sự ra đời của khái niệm tác tử gắn liền với sự xuất hiện của các khái niệm: thực thể sinh học, rô-bốt tự trị, sự mô-đun hóa trong công nghệ phần mềm. Vì đây là một lĩnh vực mới nên vẫn chưa có sự thống nhất về khái niệm tác tử. Theo [5], [7] khái niệm về tác tử có thể hiểu: đó là một thực thể vật lý hoặc ảo, tồn tại và phát triển trong môi trường sống mà nó là một thành phần trong môi trường đó. Các tác tử có khả năng hoạt động độc lập, có khả năng học hỏi và phát triển cũng như tự phân hủy trong môi trường sống. Các tác tử tương tác lẫn nhau và tương tác với môi trường sống cũng như chịu sự tác động của môi trường.

- **Hệ thống đa tác tử**

Qua khái niệm tác tử, hình ảnh MAS có thể được liên tưởng như một xã hội của những động vật bậc thấp (ví dụ đàn kiến) mà các phần tử trong đó có những nét đặc trưng như: có mối quan hệ (phụ thuộc, chi phối, ngang hàng), có tương tác trao đổi thông tin và học hỏi lẫn nhau, tác động vào môi trường và ngược lại. Theo Ferber [5], một MAS được cấu thành bởi: môi trường sống; tập các đối tượng thụ động chịu sự tác động của các tác tử lên nó; tập các tác tử là các phần tử chủ động; tập các mối quan hệ giữa các phần tử với nhau; tập các thao tác giữa các phần tử với nhau.

Với những đặc trưng của MAS, đã có nhiều mô hình phân tích thiết kế như Gaia [9], mô hình này tập trung vào việc phân tích các nhiệm vụ của các tác tử. Hoặc mô hình AALADIN [4] thì dựa trên sự tổ chức của các tác tử. Trong bài báo này, mô hình phân tích nguyên âm [13] được sử dụng. Mô hình này xem một MAS có 4 phần: tập các tác tử **A** (Agents), môi trường **E** (Environment), tập các tương tác **I** (Interactions), sự tổ chức hệ thống **O** (Organization)

Quy tắc 1: $MAS = A + E + I + O$

Quy tắc 2: Chức năng của MAS = Σ chức năng của các tác tử + chức năng phát sinh

Quy tắc 3: Trong đệ quy, một MAS có thể xem như là một tác tử (ở mức cao hơn) và ngược lại một tác tử có thể xem như là một MAS (ở mức thấp hơn). Một tác tử ở mức thấp nhất, mức 0 (không thể phân rã được) là một tác tử tồn tại thực sự gọi là tác tử nguyên tử, các tác tử ở mức khác 0 gọi là các tác tử phức hợp. Như vậy ta có: $A = A \text{ nguyên tử} \mid A \text{ phức hợp}$

- **Các tiếp cận đặc tả hình thức hệ thống đa tác tử đệ quy**

Đặc tả hình thức là sử dụng các khái niệm toán học để mô tả hệ thống, nó đặc biệt có ý nghĩa trong lĩnh vực tin học. Đặc tả hình thức dựa trên những cơ sở toán học để đặt ra các nguyên tắc, cách biểu diễn, suy diễn. Đối với MAS, đã có nhiều ngôn ngữ hình thức được sử dụng như: ngôn ngữ Z, ngôn ngữ hướng đối tượng Z [12], mạng Petri [3], DESIRE [2], logic BDI (Belief, Desire, Intention) [8], Π -calcul [10]. Mỗi ngôn ngữ thích ứng với một số mô hình của MAS, tuy nhiên với mô hình MAS đệ quy mà chúng tôi chọn lựa thì các ngôn ngữ trên chưa đáp ứng được.

3. Hệ thống đa tác tử đệ quy

3.1. Mô hình nguyên âm AEIO

Khái niệm đệ quy sử dụng trong mô hình MAS đệ quy được xây dựng dựa trên quy tắc 3 trong [13]. Trong mô hình này, sự phân rã một cách đệ quy có dạng cây, chỉ những tác tử nguyên tử ở lá (mức 0) mới tồn tại thực sự, còn những tác tử phức hợp (ở mức khác 0) là những tác tử ảo được xây dựng bằng cách gom nhóm một số tác tử ở mức cao hơn. Tất cả các tác tử phức hợp ở cùng một mức hợp thành một MAS ở mức đó.

Trước khi tìm hiểu MAS đệ quy, chúng ta đi phân tích mô hình AEIO theo 4 thành phần:

- **A và E:** giả sử rằng U là vũ trụ được tạo bởi tập các tác tử A và các đối tượng tạo nên môi trường E, ta có: $U = A \cup E$
- **I:** bao gồm tất cả các tương tác trong hệ thống. Theo [6] thì có 3 loại tương tác:
 - **Tương tác tiếp nhận (perception interaction):** là tương tác của môi trường lên các tác tử
 - **Tương tác tác động (action interaction):** là tương tác của tác tử lên môi trường
 - **Tương tác nhận thức (interaction cognitive):** là tương tác giữa các tác tử với nhau.
 Nếu cho rằng các tương tác là các hàm trong vũ trụ, và hàm này trả về các tín hiệu s (signal) thì hàm I được biểu diễn: $I: A \times U \rightarrow S$ với S là tập các tín hiệu s
 Chúng ta có thể đặc tả 3 loại tương tác trên:
 - **Tương tác tiếp nhận:** $I(A_i, e) = \text{percept}(e, A_i)$
 - **Tương tác tác động:** $I(A_i, e) = \text{action}(A_i, e)$
 - **Tương tác nhận thức:** $I(A_i, A_j) = \text{message}(A_i, A_j)$
- **O:** là sự tổ chức trong hệ thống hay chính là tập các mối quan hệ giữa các tác tử với nhau. Có 3 loại quan hệ [1]: biết (knowledge), giao tiếp (communication), lệ thuộc (subordination). O là một tập các quan hệ 2 ngôi, ứng với mỗi quan hệ thì tập các tác tử được tách thành các nhóm theo quan hệ của chúng. Ta có: $O = \{A \mathfrak{R} A\}$.

3.2. Mô hình nguyên âm AEIO đệ quy

- **Đệ quy E** **đệ quy E := e nguyên tố | {đệ quy E}**
 E có thể là một e nguyên tố hoặc là một tập hợp các E đệ quy khác. Nếu gọi E_n, E_{n-1} lần lượt là môi trường ở mức n và n-1 và P là tính chất đệ quy thì: $E_n = \{P(E_{n-1})\}$
- **Đệ quy A** **đệ quy A := A nguyên tố | {đệ quy A}**
 A có thể là một A nguyên tố hoặc là một tập hợp các A đệ quy khác. Nhưng theo định nghĩa mỗi MAS ở mức n-1 là một tác tử ở mức n. Vì thế, tính đệ quy P của A còn phụ thuộc vào cả môi trường nên: $A_n = P(\{A_{n-1}\}, \{E_{n-1}\})$
- **Đệ quy I** **đệ quy I := I nguyên tố | {đệ quy I}**
 Tính chất đệ quy P sẽ được xác minh qua 3 loại hàm I. Ở mức 0 thì: $I_0: A_0 \times U_0 \rightarrow S_0$ với S_0 là tập các tín hiệu mức 0.
 - **Tương tác tiếp nhận:** $I_0(A^i_0, e) = \text{percept}(e_0, A^i_0)$
 - **Tương tác tác động:** $I_0(A^i_0, e_0) = \text{action}(A^i_0, e_0)$
 - **Tương tác nhận thức:** $I_0(A^i_0, A^j_0) = \text{message}(A^i_0, A^j_0)$
 Ở mức n:
 - **Tương tác tiếp nhận:** $I_n(A^i_n, e_n) = \{\text{percept}(e^k_{n-1}, A^k_{n-1})\}$
 - **Tương tác tác động:** $I_n(A^i_n, e_n) = \{\text{action}(A^k_{n-1}, e^k_{n-1})\}$
 - **Tương tác nhận thức:** $I_n(A^i_n, A^j_n) = \{\text{message}(A^p_{n-1}, A^q_{n-1})\}$
- **Đệ quy O** **đệ quy O := O nguyên tố | {đệ quy O}**
 Ở mức 0 thì rõ ràng O nguyên tố chính là mối quan hệ giữa các tác tử với nhau, ở đây chỉ phân tích trong trường hợp mức của O khác 0. Bởi vì các mối quan hệ tạo nên các nhóm nên khi phân rã từ mức n xuống mức n-1, hàm O có dạng như sau:
 $O_n = A^i_n \mathfrak{R} A^j_n = \text{Gr}(A^q_{n-1}) \mathfrak{R} \text{Gr}(A^p_{n-1})$ với Gr là nhóm do A^i_n phân rã thành.

4. Áp dụng lý thuyết kiểu đặc tả hệ thống đa tác tử đệ quy

4.1. Lý thuyết kiểu

Lý thuyết kiểu là một đối tượng hấp dẫn cho các nhà tin học, toán học, sinh học từ nhiều năm nay. Đối với lĩnh vực tin học, lý thuyết kiểu là một công cụ hiệu quả để đặc tả hình thức các hệ thống, đặc biệt là các hệ thống đệ quy. Lý thuyết kiểu được xây dựng dựa trên 3 thành phần chính: toán lô-gic, λ -calcul và toán học suy diễn. Theo [11] muốn định nghĩa một kiểu nào đó, chúng ta thường đưa ra 4 luật:

- Luật khởi tạo **F** (Formation): giải thích kiểu này là gì
- Luật giới thiệu **I** (Introduction): giới thiệu các phần tử của kiểu
- Luật giản ước **E** (Elimination): từ các giả thiết, giới thiệu giản ước các phần tử của kiểu
- Luật tính toán **C** (Calculation): quy định các cách giản ước biểu thức cho đơn giản.

Vi dụ: kiểu **bool** được định nghĩa: True, False là 2 giá trị của kiểu

- Luật F:
$$\frac{}{bool \text{ is type}} (bool F)$$
- Luật I:
$$\frac{}{True : bool} (bool I_1) \quad \frac{}{False : bool} (bool I_2)$$
- Luật E:
$$\frac{tr : bool \quad c : C[True/x] \quad d : C[False/x]}{if \ tr \ then \ c \ else \ d : bool} (bool E)$$
- Luật C:
$$\frac{if \ True \ then \ c \ else \ d \rightarrow c}{if \ False \ then \ c \ else \ d \rightarrow d} (bool C)$$

Với kí hiệu phân số, thành phần trên gạch ngang là giả thiết, còn dưới gạch ngang là kết luận dựa trên những giả thiết đã cho, $C[t/x]$ là một biểu thức chứa x , biến x được thay bởi giá trị t , c : **bool** có nghĩa là c có kiểu **bool**.

4.2. Áp dụng lý thuyết kiểu đặc tả MAS đệ quy theo mô hình AEIO

- **Định nghĩa kiểu E:**

- Luật F:
$$\frac{}{E \text{ is a type}} (E F)$$
- Luật I:
$$\frac{}{e_0 : E} (E I_1) \quad \frac{(e :: l) : [E_{n-1}]}{(F_{GroupE}(e :: l)) : E_n} (E I_2)$$

Trong đó $(e :: l) : [E_{n-1}]$ là một danh sách kiểu E ở mức $n-1$ và phần tử đầu danh sách là e , tiếp theo là một dãy các phần tử l , $F_{GroupE}(l)$ là hàm nhóm các phần tử kiểu E trong danh sách l thành một phần tử kiểu E ở mức cao hơn.

- Luật E:
$$\frac{e : E \quad c : C[e_0/x] \quad f : (\forall l : [E]).(C[l/x] \Rightarrow C[F_{GroupE}l/x])}{F_{DecompE} \ e \ c \ f : C[e/x]} (E E)$$

Trong đó $(\forall l : [E]).(C[l/x] \Rightarrow C[F_{GroupE}l/x])$: với mọi danh sách l kiểu E , từ biểu thức $C[l/c]$ kéo theo $C[F_{GroupE}l/x]$, $F_{DecompE}(e \ c \ f)$ là hàm phân rã e với điều kiện $c \ f$.

- Luật C:

$$F_{DecompE} \ e_0 \ c \ f \rightarrow c$$

$$F_{DecompE} (F_{GroupE}(e :: l)) \ c \ f \rightarrow f (F_{DecompE} \ e \ c \ f) (F_{DecompE} (F_{GroupE}l) \ c \ f) (E \ C)$$

- **Định nghĩa kiểu A:**

- Luật F:
$$\frac{}{A \text{ is a type}} (A F)$$

$$\begin{aligned}
\text{- Luật I: } & \frac{}{a_0 : A} (A I_1) \quad \frac{(a :: l) : [A_{n-1}]}{(F_{GroupA} (a :: l)) : A_n} (A I_2) \\
\text{- Luật E: } & \frac{e : A \quad c : C[a_0 / x] \quad f : (\forall l : [E]). (C[l / x] \Rightarrow C[F_{GroupA} l / x])}{F_{DecompA} a c f : C[a / x]} (A E)
\end{aligned}$$

- Luật C:

$$\begin{aligned}
& F_{DecompE} e_0 c f \rightarrow c \\
& F_{DecompE} (F_{GroupE} (e :: l)) c f \rightarrow f (F_{DecompE} e c f) (F_{DecompE} (F_{GroupE} l) c f) \quad (A C)
\end{aligned}$$

• **Định nghĩa hàm I:**

I không phải là một kiểu, I là một hàm có 2 tham số, trả về tín hiệu $s: \mathbf{I} : \mathbf{A} \times \mathbf{U} \rightarrow \mathbf{S}$
 Áp dụng lý thuyết kiểu để định nghĩa đệ quy hàm I như sau:

$$I a_n u_0 \rightarrow s$$

$$I a_n F_{GroupU} (u_{m-1} :: l) \rightarrow I a_n u_m \cup I a_n F_{GroupU} l$$

• **Định nghĩa hàm O:**

O được xem là một hàm mà tham số của nó là một tác tử a, hàm trả về tập các tác tử có quan hệ với tác tử a: $\mathbf{O} : \mathbf{A} \rightarrow \mathbf{A}^n$. Áp dụng lý thuyết kiểu để định nghĩa đệ quy hàm O như

$$O a_0 \rightarrow \{a^n\}$$

sau:

$$O F_{GroupA} (a_{n-1} :: l) \rightarrow O a_{n-1} \cup O F_{GroupA} l$$

5. Ứng dụng: Đặc tả hệ thống tìm kiếm trên bản đồ địa lý bằng lý thuyết kiểu

5.1. Mô hình hóa hệ thống tìm kiếm thông tin trên bản đồ bằng mô hình AEIO

Bản đồ là như tập hợp các đối tượng địa lý có mối quan hệ với nhau. Các đối tượng có thể là điểm, đoạn thẳng, hoặc mặt bằng. Bài toán đặt ra là làm thế nào để phóng to hoặc thu nhỏ bản đồ hoặc một vùng trên bản đồ để phù hợp với việc in ấn và tìm kiếm địa điểm, đường đi... Khi phóng to một vùng, tức là chi tiết hóa các đối tượng trong vùng đó để dễ dàng tìm kiếm thì các đối tượng xung quanh phải bị thu nhỏ lại, đôi khi nó chỉ còn lại là một đối tượng đại diện cho nhóm thôi. Vì thế cần phải có cái nhìn ở mức thấp hơn (phóng to) và mức cao hơn (thu nhỏ) các đối tượng. Với yêu cầu như thế, chúng tôi áp dụng mô hình MAS đệ quy AEIO cho bài toán.

- E: tập hợp các đối tượng địa lý tạo nên môi trường: nhà hoặc đường phố
 - Mức 0: môi trường cấu thành bởi tập các đối tượng địa lý (Object Geographic- **ObGeo**) như các tòa nhà và con đường
 - Mức n: mỗi đối tượng ở mức này là tập hợp các đối tượng ở mức thấp hơn (Object Geographic Set - **ObGeoSet**). Ví dụ ở mức 1, một đối tượng có thể là «khu phố» là tập các tòa nhà và con đường ở mức 0. Ở mức 2, một đối tượng có thể là một phường/xã...
- A: tập các tác tử quản lý các đối tượng có mối quan hệ với nhau (gần nhau với một khoảng cách nhất định)
 - Mức 0: mỗi tác tử tương ứng với một hoặc một nhóm các đối tượng. Có thể là đường phố (Agent Route- **AgR**) hoặc tòa nhà (Agent Building- **AgB**)

- Mức n: mỗi tác tử ở mức này **AgBSet**, **AgRSet** là tập các tác tử ở mức thấp hơn. Ngoài ra còn có tác tử khác chứa các ràng buộc trong quá trình phân rã **AgC** (Agent Constraint).

- I: tập các thao tác
 - Mức 0: tương tác ở mức này được mô hình hóa bằng các ràng buộc di chuyển trong môi trường
 - Mức n: các tương tác chính là các tín hiệu đồng bộ giữa đối tượng và các tác tử
- O: tập các mối quan hệ
 - Mức 0: các tác tử được tổ chức thành nhóm khi chúng quản lý cùng một điểm địa lý
 - Mức n: các tác tử được tổ chức thành nhóm bởi các ràng buộc

5.2. Mô hình hóa hệ thống tìm kiếm trên bản đồ địa lý bằng lý thuyết kiểu

• Định nghĩa kiểu E:

- Luật F:
$$\frac{}{E \text{ is a type}} (EF)$$
- Luật I:
$$\frac{}{ObGeo: E} (EI_1) \quad \frac{l:[ObGeo]}{(F_{GroupE} l): E} (EI_2) \quad \frac{l:[ObGeoSet]}{(F_{GroupE} l): E} (EI_3)$$
- Luật E:
$$\frac{e: E \quad c: C[ObGeo/x] \quad f: (\forall l:[E]).(C[l/x] \Rightarrow C[F_{GroupE} l/x])}{F_{DecompE} \quad e \quad c \quad f: C[e/x]} (EE)$$
- Luật C:
$$\frac{F_{DecompE} \quad ObGeo \quad c \quad f \rightarrow c}{F_{DecompE} (F_{GroupE} (e::l)) \quad c \quad f \rightarrow f} (F_{DecompE} \quad e \quad c \quad f) (F_{DecompE} (F_{GroupE} l) \quad c \quad f) (EC)$$

• Định nghĩa kiểu A:

- ✓ Luật F:
$$\frac{}{A \text{ is a type}} (AF)$$
- ✓ Luật I:
$$\frac{}{AgR: A} (AI_1) \quad \frac{}{AgB: A} (AI_2) \quad \frac{l:[AgR]}{F_{GroupA} l: A} (AI_3)$$

$$\frac{l:[AgB]}{F_{GroupA} l: A} (AI_4) \quad \frac{l:[AgBSet]}{F_{GroupA} l: A} (AI_5) \quad \frac{l:[AgRSet]}{F_{GroupA} l: A} (AI_6)$$
- ✓ Luật E:
$$\frac{a: A \quad c: C[AgR/x] \quad f: (\forall l:[E]).(C[l/x] \Rightarrow C[F_{GroupA} l/x])}{F_{DecompA} \quad a \quad c \quad f: C[a/x]} (AE)$$

$$F_{DecompA} \quad AgR \quad c \quad f \rightarrow c$$
- ✓ Luật C:
$$F_{DecompA} \quad AgB \quad c \quad f \rightarrow c \quad (AC)$$

$$F_{DecompA} (F_{GroupA} (a::l)) \quad c \quad f \rightarrow f (F_{DecompA} \quad a \quad c \quad f) (F_{DecompA} (F_{GroupA} l) \quad c \quad f)$$

• Định nghĩa hàm I:

I là một hàm có 2 tham số, trả về kiểu tín hiệu s: $I: A \times U \rightarrow S$

$$I \quad a_n \quad u_0 \rightarrow s$$

$$I \quad a_n \quad F_{GroupU} (u_{m-1} :: l) \rightarrow I \quad a_n \quad u_m \cup I \quad a_n \quad F_{GroupU} l$$

- Định nghĩa hàm O: O được xem là một hàm mà đối số của nó là một tác tử a, hàm trả về tập các tác tử có quan hệ với tác tử a: $O: A \rightarrow A^n$

Áp dụng lý thuyết kiểu để định nghĩa đệ quy hàm O như sau:

$$O a_0 \rightarrow \{a^n\}$$

$$O F_{GroupA}(a_{n-1} :: l) \rightarrow O a_{n-1} \cup O F_{GroupA} l$$

6. Kết luận

Trong bài báo này, chúng tôi trình bày nghiên cứu mô hình hoá các hệ thống đa tác tử, đặc biệt là các hệ thống đa tác tử đệ quy. Tiếp cận đệ quy đối với hệ thống đa tác tử cho phép giảm độ phức tạp khi thiết kế hệ thống. Hướng nghiên cứu này đã và đang được phát triển, mà trong đó một trong những vấn đề đang được đặt ra: làm thế nào để đặc tả hình thức các hệ thống đa tác tử đệ quy? Trong bài báo này, chúng tôi đã đề xuất giải pháp sử dụng lý thuyết kiểu để đặc tả các hệ thống đa tác tử đệ quy. Giải pháp này mở ra những hướng phát triển: nghiên cứu khả năng cài đặt từ các đặc tả; phát triển lĩnh vực chứng minh hình thức; mở rộng giải pháp để đặc tả các loại hệ thống đa tác tử khác.

TÀI LIỆU THAM KHẢO

- [1] C. Baeijs, *Fonctionnalité Emergente dans une Société d'Agents Autonomes*, INPG, Thèse de Doctorat, France, 1998.
- [2] F. Brazier and B. D. Keplicz and N. R. Jennings and J. Treur, *Formal Specification of Multi-Agent Systems: a Real-World Case*, 1st Int. Conf. Multi-Agent Systems, Menlo Park, USA, 1995.
- [3] F. Vernadat and A. Lanusse and P. Azéma, *Modélisation par Réseaux de Petri d'un langage, Application à la Vérification de Systèmes Multi-agents*, 2èmes Journées Francophones IAD-SMA, Voiron, 1994.
- [4] J. Ferber and O. Gutknecht, A meta-model for the analysis and design of organizations in Multi-Agent Systems, *Proceedings of ICMAS'98, IEEE Computer Society*, Paris France, 1998.
- [5] J. Ferber, *Les Systèmes Multi-Agents: vers une intelligence collective*, InterEditions, 1995.
- [6] M. Ocelllo, Towards a Generic Recursive Agent Model, *Proceedings of International Conference on Artificial Intelligence (ICAI00)*, Las Vegas, USA.
- [7] M.J. Wooldridge, *Intelligent Agents, Multiagent systems: A modern approach to Distributed Artificial Intelligence*, Gerhard Weiss, MIT Press, 1999.
- [8] M. Wooldridge, *Practical Reasoning with Procedural Knowledge, Formal and Applied Practical Reasoning*, page 663-678, 1996.
- [9] M.J. Wooldridge and Jennings, N.R. and D. Kinny, *The Gaia methodology for agent-oriented analysis and design, Autonomous Agents and Multi-Agent Systems*, vol3, Kluwer Academic Publishers, 2000.
- [10] R. Milner *The polyadic pi-calculus: A tutorial, Laboratory for Foundations of Computer Science, Department of Computer Science, University of Edinburgh*, 1991.
- [11] S. Thomson, *Type Theory and Functional Programming*, Computing Laboratory, University of Kent, 1999.
- [12] V. Hilaire and A. Koukam and P. Gruer and J. P. Muller, *Formal Specification and Prototyping of Multi-agent Systems*, Lecture Notes in Computer Science, 2001.
- [13] Y. Demazeau, From Interactions to Collective Behavior in Agent-Based Systems, *European Conference on Cognitive Science*, Saint-Malo France, 1995.