

## Sử Dụng Tính Phân Cấp Để Giảm Độ Phức Tạp Trong Mô Hình Hóa Hệ Thống Sử Dụng Mạng Petri Màu

Nguyễn Thanh Tuấn<sup>1</sup>, Lê Thành Nhân<sup>2</sup> và Hoàng Thị Thanh Hà<sup>3</sup>

<sup>1</sup>Trường Đại học Công nghệ thông tin và Truyền thông Việt-Hàn

nttuan@vku.udn.vn

<sup>2</sup>Viện Công Nghệ Quốc tế Đà Nẵng

nhan.le-thanh@unice.fr

<sup>3</sup>Trường Đại học Kinh tế

ha.htt@due.udn.vn

Đại học Đà Nẵng, Việt Nam

**Tóm Tắt.** Trong phát triển phần mềm, việc mô hình hóa và kiểm chứng phần mềm đóng vai trò quan trọng, nhất là trong các hệ thống phần mềm ứng dụng doanh nghiệp. Trong nhiều nghiên cứu gần đây, mạng Petri màu, với cơ sở lý thuyết vững chắc, nổi lên như là một công cụ hiệu quả trong mô hình hóa và kiểm chứng phần mềm. Sự phân cấp là một khái niệm quan trọng trong mạng Petri màu, cho phép tổ chức mạng Petri màu dưới dạng tập hợp các mô hình con, tương tự như các chương trình phần mềm được cấu trúc thành các mô đun. Theo cách này, một mô hình có thể được định nghĩa một lần và tái sử dụng nhiều lần. Trong bài báo này chúng tôi sử dụng mạng Petri màu phân cấp để mô hình hóa tiến trình nghiệp vụ của hệ thống Cửa hàng trực tuyến, đồng thời phát triển PyCPN, một công cụ mô hình hóa sử dụng mạng Petri màu phân cấp, để mô tả mô hình.

**Từ Khóa:** Mạng Petri màu, Mạng Petri màu phân cấp, Mô hình hóa.

**Abstract.** In software development, modeling and validation play an important role, especially in enterprise application software systems. In many recent studies, the Colored Petri Nets, with its strong theoretical base, has emerged as an effective tool in software modeling and validation. Hierarchy is an important concept in Colored Petri Net, which allows the organization of a Colored Petri Net as a set of sub-models, similar to software programs structured into modules. In this way, a model can be defined once and reused. In this paper we use a Hierarchical Colored Petri Net to model an Online Store system, and develop PyCPN, a modeling tool using for the Hierarchical Colored Petri Net, to describe the model.

**Keyword:** Colored Petri Nets, Hierarchical Colored Petri Nets, Modeling

### 1 Mở đầu

Mô hình hóa các hệ thống hiện đại đưa ra rất nhiều thách thức mà một số thách thức có thể được giải quyết bằng cách sử dụng các mô hình thích hợp. Một kỹ thuật nổi tiếng để mô hình hóa hệ thống là sử dụng mạng Petri. Mạng Petri cung cấp một ngôn ngữ đồ họa để xây dựng các hệ thống một cách trực quan và chính xác. Các mô hình này có thể được sử dụng để mô phỏng, phân tích các thuộc tính và đánh giá hệ thống. Điều này cho phép phát hiện các lỗi có thể xảy ra trong hệ thống ở các giai đoạn sớm và từ đó làm giảm chi phí phát triển tổng thể. Kích thước ngày càng tăng của các hệ thống dẫn đến việc các mô hình trở nên cồng kềnh và khó kiểm soát.

Mạng Petri phân cấp sử dụng mô hình cây phân cấp chia nhỏ một hệ thống lớn thành một số hệ thống con, từ đó làm giảm độ phức tạp của hệ thống lớn. Điều này giúp người dùng tập trung vào một phần hệ thống con cụ thể mà không cần phải tính đến toàn bộ hệ thống. Ngoài ra, các hệ thống con có thể được tái sử dụng lại một cách dễ dàng tại nhiều vị trí trong cùng một hệ thống hoặc thậm chí có thể được sử dụng trong một hệ thống khác nếu có nhu cầu tương tự.

Bài báo này đề xuất sử dụng mạng Petri màu phân cấp (Hierarchical Colored Petri Net HCPN) để mô hình hóa hệ thống, ở đây hệ thống của hàng trực tuyến được sử dụng để minh họa cho cách tiếp cận này, chúng tôi cũng đã phát triển thư viện PyCPN với mục đích mô tả và thực thi các mô hình HCPN. Phần còn lại của bài viết này được tổ chức như sau: Phần 2 trình bày một số nghiên cứu liên quan, phần 3 trình bày định nghĩa hình thức của mạng Petri màu và mạng Petri màu phân cấp. Sau đó, bài toán mô hình hóa hoạt động của hệ thống Cửa hàng trực tuyến bằng HCPN được trình bày trong Phần 4. Phần kết luận được đưa ra trong Phần 5.

## 2 Các nghiên cứu liên quan

Từ năm 1991, hệ thống phân cấp trong mạng Petri [4] đã được trình bày trong nhiều bài báo. Trong bài báo [3], công cụ HiPS cung cấp các chức năng cơ bản, giao diện người dùng đồ họa, khả năng mô tả các mạng Petri phân cấp và một bộ kiểm tra mô hình logic thời gian tuyến tính. Trong [1], các tác giả xác định hệ thống phân cấp vị trí với các cung khác nhau làm cung ức chế, đặt lại cung và chuyển cung tuân theo thứ bậc này. Tổng thứ tự các vị trí mã hóa hệ thống phân cấp tùy thuộc vào các loại vòng cung mà các phần mở rộng khác nhau được đưa ra. Bên cạnh đó, mạng Petri phân cấp dựa trên sự thay thế chuyển tiếp như ví dụ trong [5] hay dựa trên sự thay thế vị trí và mạng Petri hướng đối tượng [9,10] đã được xem xét. CPN [2] cũng hỗ trợ các vị trí hợp nhất bằng cách xác định các tập hợp các vị trí giống hệt nhau về mặt chức năng. Phương pháp tiếp cận Net-in-net [15,16] cũng như mạng đối tượng [7,8] đều bổ sung các thành phần phân cấp cho mạng Petri.

Mạng Petri màu phân cấp cũng đã được sử dụng trong mô hình hóa các quy trình nghiệp vụ [18]. Trong [19], các tác giả đã đề xuất mô hình sử dụng mạng Petri phân cấp trong mô hình hóa và xác thực quy trình nghiệp vụ kết hợp với ngôn ngữ luật ECA.

Nhiều công cụ đã được phát triển để phục vụ mô hình hóa, mô phỏng và xác thực mạng Petri màu. Tiêu biểu là CPNTool [2,13], đây là công cụ phổ biến để mô hình hóa và mô phỏng mạng Petri màu. Sử dụng giao diện người dùng đồ họa, CPNtools có tính năng kiểm tra cú pháp, tạo mã và phân tích không gian trạng thái, công cụ này không phải mã nguồn mở nên hạn chế trong việc sinh mã cũng như tích hợp với các công cụ khác.

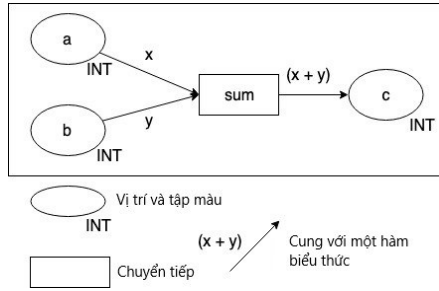
Snoopy [14] là một nền tảng mạng Petri thống nhất với giao diện người dùng đồ họa. Nó cho phép mô hình và mô phỏng các mạng Petri có màu và không có màu thuộc các lớp khác nhau, hỗ trợ phân tích và cấu trúc phân cấp của các mô hình.

SNAKES [11] là một thư viện mạng Petri, chủ yếu dành cho ngôn ngữ Python nhưng có thể chuyển sang các ngôn ngữ khác. Nó định nghĩa một biến thể chung của mạng Petri, với màu là kiểu dữ liệu của Python, có thể được tạo và thao tác thông qua thư viện, cũng như được thực thi để khám phá không gian trạng thái. Hiện tại, ZINC [12], là một phiên bản tiếp theo của SNAKES, hướng đến sự đơn giản đang được phát triển. Điều chỉnh của SNAKES hay ZINC đó là thiếu khả năng mô tả các mạng Petri phân cấp, mà điều này sẽ được bổ sung, phát triển trong thư viện PyCPN do chúng tôi xây dựng.

## 3 Mạng Petri màu

### 3.1 Mạng Petri màu

Một mạng Petri (Petri Nets) [6] là một dạng đồ thị được tạo ra từ một số kiểu đối tượng là: các vị trí, các chuyển tiếp, các cung và các token. Các cung kết nối trực tiếp từ các vị trí đến các chuyển tiếp hoặc ngược lại. Một chuyển tiếp được kích hoạt khi mỗi vị trí trong điều kiện bắt buộc của việc chuyển tiếp được thỏa mãn. CPN là phần mở rộng về mặt khái niệm của mạng Petri. CPN giữ lại các đặc tính hữu ích của mạng Petri, đồng thời mở rộng định nghĩa hình thức của mạng Petri để cho phép đưa ra nhiều kiểu token. Một ví dụ về CPN được thể hiện trực quan trong Hình 1.



Hình. 1. Một ví dụ về CPN

**Định nghĩa 1.** Mạng Petri màu (CPN - Coloured Petri Net) là một tập hợp  $N = (P, T, A, \Sigma, C, N, E, G, I)$ , mà ở đây:

- $P$  là tập các vị trí
- $T$  là tập các chuyển tiếp
- $A$  là tập các cung
- $\Sigma$  là tập màu định nghĩa trong mô hình CPN. Tập này bao gồm tất cả các màu có thể, các phương thức và các hàm sử dụng trong CPN.
- $C$  là một hàm các màu. Ánh xạ các vị trí trong  $P$  với các màu trong  $\Sigma$ .
- $N$  là một hàm nút. Ánh xạ tập  $A$  với  $(P \times T) \cup (T \times P)$ .
- $E$  là một hàm biểu thức cung. Ánh xạ mỗi cung với biểu thức  $e$ . Kiểu của đầu vào và đầu ra của biểu thức cung phải tương ứng với kiểu tại các nút mà đường cung kết nối. Sử dụng hàm nút và hàm biểu thức cung cho phép nhiều cung kết nối cùng một cặp nút với các biểu thức cung khác nhau.
- $G$  là một hàm canh. Kết nối mỗi chuyển đổi  $t \in T$  với một biểu thức canh  $g$ . Đầu ra của biểu thức canh là giá trị Boolean: đúng hoặc sai.
- $I$  là một hàm khởi tạo. Kết nối mỗi vị trí  $p$  với một biểu thức khởi tạo  $i$ . Biểu thức khởi tạo tương ứng với một đa tập các token với màu sắc tương ứng với màu tại vị trí  $C(p)$ .

Sự phân cấp là một khái niệm quan trọng trong CPN, cho phép tổ chức CPN dưới dạng tập hợp các mô hình con, tương tự như trong đó các hệ thống phân mềm được cấu trúc với các mô đun. Theo cách này, một mô hình có thể được định nghĩa một lần và tái sử dụng nhiều lần.

**Định nghĩa 2.** Một mô đun CPN được định nghĩa như một dãy  $\Sigma_M = (\Sigma, T_{sub}, P_{port}, PT)$ , ở đây:

- $\Sigma = (P, T, A, C, V, W, G, E, I)$  là một CPN;
- $T_{sub} \subseteq T$  là tập các Chuyển tiếp thay thế;
- $P_{port} \subseteq P$  là tập các Vị trí cổng (Port);
- $PT: P_{port} \rightarrow \{IN, OUT, I/O\}$  là một hàm kiểu Port gắn mỗi kiểu Port với mỗi Vị trí Port.

**Định nghĩa 3.** Một mạng Petri phân cấp (Hierarchical Colored Petri Net - HCPN) được mô tả như một dãy  $\Sigma_H = (S, SM, PS, FS)$ , ở đây:

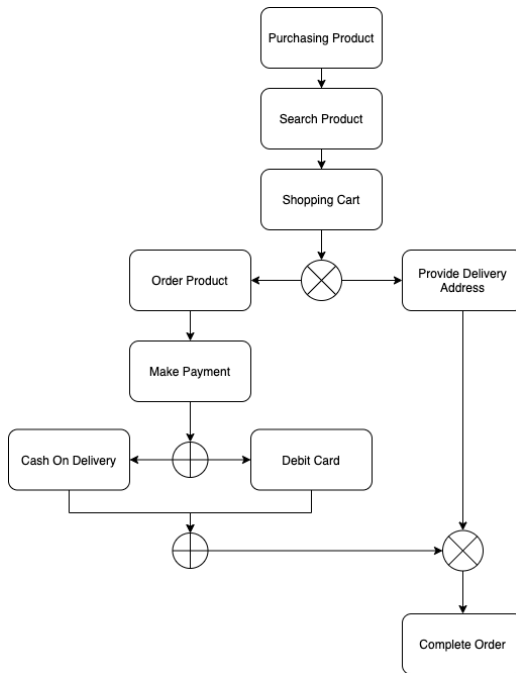
- $S$  là tập hữu hạn các mô đun. Mỗi mô đun là một mô đun
- $$CPNs = ((P^s, T^s, A^s, C^s, V^s, W^s, G^s, E^s, I^s), T_{sub}^s, P_{port}^s, PT^s),$$
- với yêu cầu  $(P^{s_1} \cup T^{s_1}) \cap (P^{s_2} \cup T^{s_2}) = \emptyset$  với mọi  $s_1, s_2 \in S$  sao cho  $s_1 \neq s_2$
- $SM: T_{sub} \rightarrow S$  làm một hàm mô đun con gắn mỗi mô đun con với mỗi Chuyển tiếp thay thế.
  - $PS$  làm một hàm quan hệ port-socket gắn một quan hệ port-socket  $PS(t) \subseteq P_{sock}(t) \times P_{port}^{SM(t)}$  với mỗi Chuyển tiếp thay thế  $t$ , với yêu cầu  $ST(p) = PT(p'), C(p) = C(p')$ , and  $I(p)\langle \rangle = I(p')\langle \rangle$  với mọi  $(p, p') \in PS(t)$  và mọi  $t \in T_{sub}$ .

- $FS \subseteq 2^P$  làm một tập các tập liên hợp không rỗng sao cho  $C(p) = C(p')$  and  $I(p)\langle \rangle = I(p')\langle \rangle$  với mọi  $p, p' \in fs$  với mọi  $fs \in FS$ .

## 4 Mô hình hóa hệ thống sử dụng mạng Petri màu phân cấp

### 4.1 Trường hợp nghiên cứu: Hệ thống quản lý Cửa hàng trực tuyến

Một hệ thống Cửa hàng trực tuyến được phát triển với mục tiêu là giới thiệu sản phẩm và bán hàng và giao sản phẩm đúng hạn. Thanh toán có thể được thực hiện online bằng thẻ ghi nợ/ ngân hàng điện tử hoặc bằng tiền mặt khi nhận hàng.



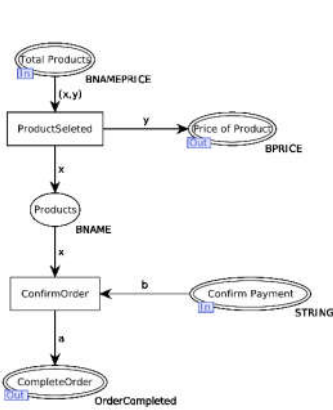
Hình. 2. Tiến trình nghiệp vụ bán hàng của cửa hàng trực tuyến

Mô hình Mua hàng đã được thể hiện trong Hình 2 cho thấy các hành động của khách hàng khi mua hàng trực tuyến. Hoạt động mua hàng được phân tách thành các hoạt động phụ như: Tìm kiếm, Cập nhật giỏ hàng, Đặt mua, Thanh toán, Cung cấp địa chỉ giao hàng và các dịch vụ web sẽ được phát triển cho từng hoạt động phụ này.

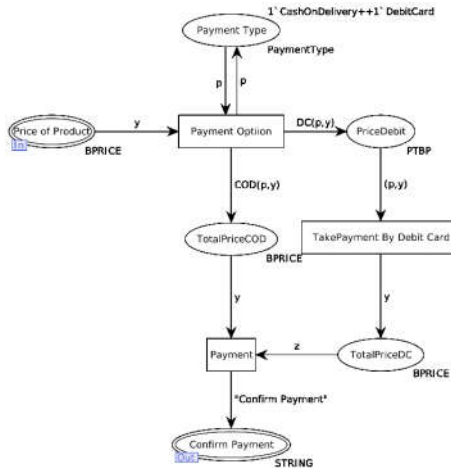
Trong Hình 2, các dịch vụ Tìm kiếm và Giỏ hàng được trình bày tuần tự, trong khi đặt mua sản phẩm và Cung cấp địa chỉ giao hàng được thực hiện đồng thời. Hình thức thanh toán thì chỉ được chọn một trong hai hình thức là thanh toán bằng tiền mặt hoặc thanh toán trực tuyến.

### 4.2 Mô hình hóa hoạt động của Cửa hàng trực tuyến bằng mạng Petri màu phân cấp của

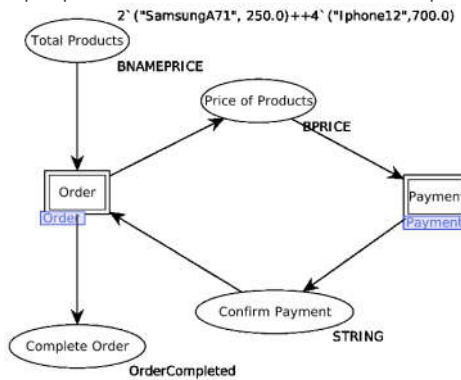
Trong phần này, chúng tôi áp dụng mô hình CPN cho hệ thống hoạt động Cửa hàng trực tuyến. Sự phụ thuộc phân cấp giữa các mô đun con sẽ được kết nối bằng các socket. Hình 3 thể hiện bức tranh tổng thể của hệ thống phân cấp này.



Hình 3a. Mô đun dịch vụ Order



Hình 3b. Mô đun dịch vụ Payment



Hình 3c. Mô đun chính là mô hình phân cấp

Hình 3. Mô hình CPN Cửa hàng trực tuyến

Trong hình 3c, có hai Chuyển tiếp Order và Payment, các vị trí đầu vào và đầu ra từ các Chuyển tiếp thay thế được đặt tại các socket Input, Output hoặc Input/Output. Ở đây chuyển tiếp Order gửi quyền kiểm soát đến chuyển tiếp Payment qua socket PriceOfProduct và nhận quyền kiểm soát từ chuyển tiếp Payment thông qua socket ConfirmPayment.

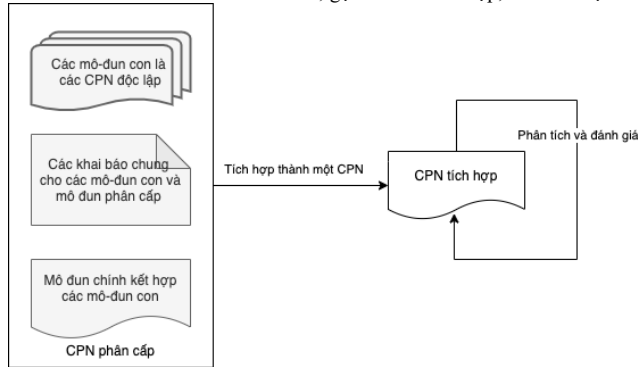
Chi tiết về mô đun Order và Payment được mô tả trong Hình 3a và 3b. Trong Hình 3c, Vị trí TotalProduct và ConfirmPayment là socket Input và PriceOfProduct và CompleteOrder là socket Output.

### 4.3 Xây dựng thư viện PyCPN

Để cài đặt mô hình CPN phân cấp chúng tôi đã phát triển thư viện **PyCPN**. Đây là thư viện sử dụng trong mô hình hóa và xác thực các CPN, được xây dựng trên nền tảng ZINC[12]. ZINC cho phép tạo CPN từ chương trình Python hoặc từ một định dạng thống nhất đơn giản, sau đó biên dịch sang các ngôn ngữ Go, Python, CoffeeScript để thực thi.

Ngoài các chức năng thừa hưởng từ ZINC như là: sử dụng các ngôn ngữ lập trình Python, Go, CoffeeScript trong khai báo token màu, hỗ trợ các cung được chú thích bởi các giá trị, biến, biểu thức.... **PyCPN** còn cho phép xây dựng các CPN phân cấp, kết hợp giữa các mô đun con độc lập và mô đun

chính. Điều này có ý nghĩa quan trọng trong việc tích hợp và tái sử dụng các mô hình đã xây dựng. Mô hình CPN phân cấp này, cùng chia sẻ chung một tập các khai báo giữa các mô đun, có thể tạo thành một CPN, khi kết hợp tất cả các mô đun con và mô đun chính, gọi là CPN tích hợp, như thể hiện trong Hình 4.



Hình 4. CPN tích hợp từ CPN phân cấp

Việc phân tích, đánh giá mô hình CPN phân cấp có thể được thực hiện thông qua phân tích, đánh giá CPN tích hợp này.

#### 4.4 Cài đặt và thử nghiệm mô hình

Sử dụng PyCPN, mô hình hoạt động của Cửa hàng trực tuyến được mô tả như trong Hình 5.

```

lang python
module Order, Payment
hcpn "OnlineStore" :
  place TotalProduct BNAMEPRICE = ('SamsungA71',250.0), ('Iphone12',500.0)
  place PriceOfProduct BPRICE
  place ConfirmPayment str
  place CompleteOrder OrderCompleted
  subtrans Order:
    < TotalProduct
    < ConfirmPayment
    > PriceOfProduct
    > CompleteOrder
  IN TotalProduct
  IN ConfirmPayment
  OUT PriceOfProduct
  OUT CompleteOrder
  subtrans Payment:
    < PriceOfProduct
    > ConfirmPayment
  IN PriceOfProduct
  OUT ConfirmPayment
    
```

Hình 5. Mô tả CPN phân cấp là mô đun chính của mô hình hoạt động Cửa hàng trực tuyến

Mô hình này kết hợp với 2 mô đun con là Order và Payment, được mô tả như trong Hình 6 và 7.

```

lang python
cpn "Order":
  place TotalProduct BNAMEPRICE
  place Product BNAME
  place PriceOfProduct BPRICE
  place ConfirmPayment str
  trans ProductSelected:
    < TotalProduct (var,var) = (x,y)
    
```

```

> Product var = x
> PriceOfProduct var = y
trans ConfirmOrder
< Product var = x
< ConfirmPayment var = b
> CompleteOrder var = a

```

**Hình 6.** Mô tả mô đun con Order

```

lang python
cpn "Payment":
  place PaymentType PaymentType
  place PriceDebit PTBP
  place TotalPriceCOD BPRICE
  place TotalPriceDC BPRICE
  place PriceOfProduct BPRICE
  place ConfirmPayment str
  trans PaymentOption:
    < PriceOfProduct var = y
    < PaymentType var = p
    > PaymentType var = p
    > PriceDebit expr = y if p == 'CashOnDelivery' else 0.0
    > TotalPriceCOD expr = y if p == 'DebitCard' else 0.0
  trans TakePaymentByDebitCard:
    < PriceDebit (var,var) = (p,y)
    > TotalPriceDC var = y
  trans Payment:
    < TotalPriceCOD var = y
    < TotalPriceDC var = z
    > ConfirmPayment val = "Confirm Payment"

```

**Hình 7.** Mô tả mô đun con Payment

Mô đun chính và các mô đun con đều sử dụng chung một khai báo như trong Hình 8.

```

declare """
@param BNAME: Product name
@type BNAME: `str`
@param BPRICE: Product price
@type BPRICE: `float`
@param OrderCompleted: Order Completed message
@type OrderCompleted: `str`
@param PaymentType: Payment Type
@type PaymentType: `list('CashOnDelivery','DebitCard')`
@param BNAMEPRICE: NAMExPRICE
@type BNAMEPRICE: `(str,float)`
@param PTBP: PaymentTypexPRICE
@type PTBP: `(str,float)`
@param BNAME: Product name
@type BNAME: `str`
"""

```

**Hình 8.** Khai báo chung cho CPN phân cấp Cửa hàng trực tuyến

## 5 Kết luận

Bài báo này đã đề xuất xây dựng mô hình hoạt động của Cửa hàng trực tuyến sử dụng mạng Petri màu phân cấp, và thông qua phân tích không gian trạng thái mạng Petri này, ta có thể xác minh tính đúng đắn và độ tin cậy của mô hình Cửa hàng trực tuyến trước giai đoạn triển khai phát triển thực tế. Mô hình được khai báo và cài đặt bằng thư viện PyCPN. Việc dịch mô hình thành một thư viện Python để có thể sử

dụng như một phần của logic nghiệp vụ của hệ thống phần mềm khi triển khai phát triển thực tế sẽ là hướng nghiên cứu tiếp theo của chúng tôi.

### Tài liệu tham khảo

1. S. Akshay et al. “On petri nets with hierarchical special arcs”. In: Leibniz International Proceedings in Informatics, LIPICs 85 (2017). issn: 18688969.
2. CPNTools. url: <http://cpntools.org/> (visited on 09/20/2020).
3. Yojiro Harie and Katsumi Wasaki. “A petri net design and verification platform based on the scalable and parallel architecture: HIPS”. In: Advances in Intelligent Systems and Computing 558 (2018), pp. 265–273. issn: 21945357
4. Peter Huber, Kurt Jensen, and Robert M. Shapiro. “Hierarchies in coloured petri nets”. In: (1991), pp. 313–341.
5. Kurt Jensen. “High-level Petri Nets: Theory and Application”. In: (2011), pp. 1–26.
6. Kurt Jensen and Lars M. Kristensen. Coloured Petri Nets: Modelling and validation of concurrent systems. 2009, pp. 1–384. isbn: 9783642002830.
7. Michael K“ohler-Bußmeier and Frank Heitmann. “Liveness of safe object nets”. In: Fundamenta Informaticae 112.1 (2011), pp. 73–87. issn: 01692968.
8. Michael K“ohler-Bußmeier and Frank Heitmann. “Safeness for object nets”. In: Fundamenta Informaticae 101.1-2 (2010), pp. 29–43. issn: 01692968. doi: 10.3233/FI-2010-273.
9. Charles Lakos. “Object Oriented Modelling with Object Petri Nets”. In: (2001), pp. 1–37.
10. Toshiyuki Miyamoto and Sadatoshi Kumagai. “A survey of object-oriented petri nets and analysis methods”. In: IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E88-A.11 (2005), pp. 2964–2971. issn: 17451337.
11. Franck Pommereau. “SNAKES: A flexible high-level Petri nets library”. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9115 (2015), pp. 254–265. issn: 16113349.
12. Franck Pommereau. ZINC: a compiler for "any language"-coloured Petri nets. Tech. rep. IBISC, University of Evry, University of Paris-Saclay, 2018.
13. Anne Vinter Ratzter et al. “CPN tools for editing, simulating, and analysing Coloured Petri Nets”. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 2679 (2003), pp. 450–462. issn: 16113349.
14. Heiner, Monika & Herajy, Mostafa & Liu, Fei & Rohr, Christian & Schwarick, Martin. (2012). Snoopy – A Unifying Petri Net Tool. Application and Theory of Petri Nets. 398-407. 10.1007/978-3-642-31131-4\_22.
15. Rudiger Valk. “Object Petri Nets”. In: (2004), pp. 819–848. doi: 10.1007/978-3-540-27755-2\_23.
16. Rudiger Valk. “Petri nets as token objects an introduction to elementary object nets”. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 1420 (1998), pp. 1–24. issn: 16113349.
17. Aalst, Wil & Stahl, Christian. (2011). Modeling Business Processes: A Petri Net Oriented Approach. 10.7551/mitpress/8811.001.0001.
18. N. T. Tuan, L. T. Nhan and H. T. Thanh Ha, "Toward a Model for Verification of Business Logic Layer in 3-Layer Architecture: CPN-ECA Model," 2019 11th International Conference on Knowledge and Systems Engineering (KSE), Da Nang, Vietnam, 2019, pp. 1-6.