ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

NGUYỄN TRỊNH ĐÔNG

KIỂM CHỨNG HÌNH THỨC CÁC HỆ THỐNG THỜI GIAN THỰC HƯỚNG THÀNH PHẦN BẰNG KĨ THUẬT MODEL-CHECKING

Chuyên ngành : Kỹ thuật Phần mềm

Mã số : 62.48.01.03

TÓM TẮT LUẬN ÁN TIẾN SĨ NGÀNH CÔNG NGHỆ THÔNG TIN

Hà Nội – 2017

Công trình được hoàn thành tại:

Trường Đại học Công nghệ - Đại học Quốc gia Hà Nội.

Người hướng dẫn khoa học: - TS Đặng Văn Hưng

- PGS.TS. Trương Anh Hoàng

Phản biện 1: .....

Phản biện 2: .....

Phản biện 2: .....

Luận án tiến sĩ sẽ được bảo vệ trước hội đồng cấp Đại học Quốc gia chấm luận án tiến sĩ họp tại.....

Vào hồi giờ ngày tháng năm

Có thể tìm hiểu luận án tại:

- Thư viện Quốc gia Việt Nam
- Trung tâm Thông tin – Thư viện, Đại học Quốc gia Hà Nội

Chương 1

Giới thiệu

1.1 Đặt vấn đề

Phát triển phần mềm thời gian thực dựa trên thành phần đóng một vai trò quan trọng trong lĩnh vực công nghệ phần mềm. Tuy nhiên, các phương pháp đã đề xuất vẫn còn hạn chế như thiếu mô hình thành phần cho thành phần phần mềm thời gian thực, làm thế nào để có thể đặc tả, phân tích đánh giá, và kiểm chứng sự tương tác giữa các thành phần phần mềm với nhau và với môi trường của chúng trên khía cạnh chức năng và phi chức năng. Luận án đã nghiên cứu và đề xuất các kỹ thuật để giải quyết bài toán trên. Thứ nhất, luận án đề xuất mô hình cho thành phần phần mềm thời gian thực với thể thức tương tác tương tranh thời gian (timed concurrent interaction protocol), và bổ sung đặc tả tài nguyên vào thể thức tương tác tương tranh nhằm lập luận cho tính sử dụng hiệu quả nguồn tài nguyên của hệ thống. Thứ hai, luận án đề xuất mở rộng lý thuyết giao diện thành phần trở thành lý thuyết giao diện thời gian thực nhằm mô hình hóa và đặc tả các hoạt động của hệ thống thời gian thực dựa trên thành phần. Thứ ba, luận án áp dụng lý thuyết "tính đúng đắn bởi cách xây dựng" và lý thuyết "thiết kế bằng hợp đồng" để đề xuất một kỹ thuật đặc tả và mô hình hóa hệ thống thời gian thực bằng hợp đồng thời gian và hợp đồng thời gian ràng buộc tài nguyên nhằm phân tích, đánh giá hệ thống thời gian thực dựa trên hợp đồng trên hai khía cạnh chức năng và phi chức năng. Bên cạnh đó, luận án cũng đề xuất ngôn ngữ đặc tả thời gian thực như một sự gia tăng tính khả thi của lý thuyết được đề xuất.

1.2 Các kết quả chính của luận án

Luận án có ba kết quả chính được trình bày như sau. Luận án mở rộng mô hình PECOS (PErvasive COmponent Systems) cho thành phần phần mềm thời gian thực với mục đích tạo ra một mô hình có thể áp dụng cho nhiều hệ thống có nền tảng phần cứng khác nhau. Luận án đề xuất mô hình cho thành phần phần mềm thời gian thực với thể thức tương tác tương tranh thời gian (timed concurrent interaction protocol), và bổ sung đặc tả tài nguyên vào thể thức tương tác tương tranh nhằm lập luận cho tính sử dụng hiệu quả nguồn tài nguyên của hệ thống.

Luận án mở rộng lý thuyết giao diện thành phần với các ràng buộc thời gian để đặc tả và mô hình hóa các thành phần phần mềm thời gian thực. Luận án sử

dụng ôtômát thời khoảng để biểu diễn hữu hạn các dãy hành vi của giao diện thành phần và môi trường thời gian thực nhằm mục đích phân tích, đánh giá các khía cạnh như tính chất làm mịn, các phép ghép và phép cắm giữa môi trường vào thành phần phần mềm.

Luận án đề xuất kỹ thuật cải tiến đặc tả thành phần phần mềm bằng hợp đồng thời gian và hợp đồng thời gian với các ràng buộc tài nguyên nhằm phân tích và đánh giá đầy đủ các tính chất của thành phần phần mềm trên hai khía cạnh chức năng và phi chức năng. Do đó, thành phần phần mềm thời gian thực được thiết kế sẽ được xem xét một cách đầy đủ. Luận án cũng đề xuất ngôn ngữ đặc tả thời gian thực mẫu nhằm hợp nhất các ngôn ngữ đặc tả thời gian thực với đầy đủ tính năng đặc tả về phần chức năng và phi chức năng cho thành phần phần mềm. Bằng cách này, các thành phần phần mềm thời gian thực được đặc tả đầy đủ và làm cơ sở cho cho việc nâng cao chất lượng phần mềm. Tất cả các đóng góp trong luận án hướng đến việc tìm ra những kỹ thuật phân tích, đánh giá và kiểm chứng hệ thống thời gian thực dựa trên thành phần.

Các kết quả nghiên cứu trong luận án có mối liên hệ chặt chẽ với nhau trong việc tích hợp tạo thành một giải pháp hoàn chỉnh từ việc đề xuất mô hình phần mềm thời gian thực đến thể thức tương tác tương tranh trong mô hình và các kỹ thuật đặc tả và mô hình hóa hệ thống thời gian thực bằng lý thuyết giao diện và hợp đồng thời gian thực cho đến đề xuất ngôn ngữ đặc tả thời gian thực mẫu. Như vậy, kết quả trong luận án hướng đến một giải pháp đầy đủ cho việc đặc tả, mô hình hóa và kiểm chứng tính đúng đắn hệ thống thời gian thực dựa trên thành phần.

1.3 Bố cục của luận án

Các nội dung còn lại của luận án được tổ chức như sau: Chương 2 trình bày các kiến thức cơ bản cũng như các khái niệm về phát triển hệ thống dựa trên thành phần. Chương 3 trình bày mô hình thành phần phần mềm và thể thức tương tác tương tranh trong các thành phần phần mềm thời gian thực, cùng với các thuật toán kiểm chứng sự tuân thủ của dãy hành vi của môi trường hệ thống với thể thức tương tác của thành phần phần mềm trên hai khía cạnh chức năng và phi chức năng. Chương 4 trình bày kỹ thuật đặc tả giao diện thành phần phần mềm thời gian thực trên quan hệ giữa đầu vào và đầu ra của thành phần phần mềm. Chương 5 trình bày đặc tả các dịch vụ và thành phần phần mềm thời gian thực với các thiết kế thời gian. Chương 6 trình bày kỹ thuật đặc tả yếu tố chức năng và phi chức năng nhằm phân tích và đánh giá chất lượng dịch vụ dựa trên các ràng buộc phi chức năng, đồng thời ước lượng tài nguyên hệ thống tiêu thụ và sử dụng trong quá trình hoạt động ngay từ giai đoạn đầu của sự phát triển hệ thống phần mềm. Chương 7, luận án tổng kết các kết quả nghiên cứu và hướng phát triển tiếp theo.

Chương 2

Các kiến thức cơ bản

2.1 Phần mềm hướng thành phần

2.1.1 Kỹ nghệ phần mềm dựa trên thành phần

Phát triển phần mềm dựa trên thành phần có một số dặc điểm sau:

(i) Là một đơn vị phần mềm độc lập, thực thi một chức năng đầy đủ nào đó.
(ii) Có tính ẩn đối với hệ thống bên ngoài. Một thành phần phần mềm chỉ được quan sát thông qua giao diện của chúng. Khi tương tác với các thành phần khác thông qua một thỏa thuận gọi là hợp đồng.
(iii) Là chủ thể cho nhiều mục đích sử dụng khác nhau. Một thành phần phần mềm được thiết kế và cài đặt sao cho chức năng của thành phần đó được tái sử dụng và phục vụ nhiều hệ thống khác nhau.
(iv) Thành phần phần mềm có thể được tái sử dụng và có thể thay thế bằng thành phần phần mềm khác.

2.1.2 Tính đúng đắn bởi cách xây dựng

Là kỹ thuật phát triển phần mềm dựa trên việc đặc tả các thành phần phần mềm sao cho nếu các thành phần phần mềm ghép được với nhau thì hệ thống không còn lỗi, phương pháp tiếp cận này là biến thành phần phần mềm về một dạng mô hình toán học để chứng minh.

2.1.3 Kiến trúc hệ thống phần mềm dựa trên thành phần

2.2 Hệ thống thời gian thực

Theo tổng kết của tác giả M. Timmerman và L. Perneel cho thấy có nhiều cách định nghĩa hệ thống thời gian thực nhưng theo các tác giả hệ thống thời gian thực là hệ thống: *" . . . phản ứng theo cách đoán trước được (kịp thời) các kích thích không dự đoán được đến từ bên ngoài."*

2.2.1 Ôtômát thời gian

Mọi hoạt động trong thế giới chúng ta đều phụ thuộc vào thời gian, ở một mức hoạt động nào đó yếu tố thời gian có được xem xét hoặc ẩn đi tùy thuộc vào bài toán cần giải quyết.

Định nghĩa 2.1 (Ôtômát thời gian)**.** *Ôtômát thời gian là một bộ* $M = \langle L, \Sigma, \ell_0, \mathbb{C}, T, \Gamma \rangle$*, trong đó* L *là tập vị trí,* Σ *là bảng chữ cái,* ℓ_0 *là vị trí khởi tạo,* $\mathbb{C}$ *là tập các đồng hồ,* $T : L \times \Sigma \times \Phi(\mathbb{C}) \times 2^{\mathbb{C}} \times L$ *là bảng chuyển, và* Γ *là tập các vị trí có thể chấp nhận được.*

Sự dịch chuyển của ôtômát thời gian có hai hình thức.

(i) Thời gian trôi: $\langle \ell, \nu \rangle \xrightarrow{d} \langle \ell, \nu + d \rangle$ với mọi $d \in \mathbb{R}_{\geq 0}$.
(ii) Dịch chuyển rời rạc: $\langle \ell, \nu \rangle \xrightarrow{e} \langle \ell', \nu' \rangle$ nếu tồn tại một dịch chuyển $e = \langle \ell, a, \phi, \hbar, \ell' \rangle \in T$ sao cho $\nu \models \phi$, $\phi \in \Phi(\complement)$, và $\nu' = [\hbar \mapsto 0]\nu$.

2.2.2 Ôtômát trọng số

Ôtômát trọng số là sự mở rộng của ôtômát thời gian nhưng cho phép tính toán chi phí cho từng hoạt động của hệ thống thời gian thực.

2.3 Kiểm chứng hệ thống thời gian thực

2.3.1 Đặc tả và mô hình hóa hệ thống

Đặc tả và mô hình hóa là một trong những bước quan trọng trong kiểm chứng phần mềm. Hiện nay, các kỹ thuật có nhiều ưu điểm trong đặc tả thành phần phần mềm là sử dụng lý thuyết giao diện và lý thuyết hợp đồng. Để mô hình hóa các hệ thống thời gian thực, người ta sử dụng ôtômát thời gian hoặc các kỹ thuật tương đương để làm mô hình hệ thống.

2.3.2 Đặc tả tính chất của hệ thống

Khi kiểm chứng, các tính chất của hệ thống thời gian thực được đặc tả bằng các hệ logic thời gian. Phương pháp đầu tiên là dùng hệ toán mệnh đề. Hệ toán mệnh đề đã mệnh đề hóa các phát biểu bằng các ký hiệu từ đó kết hợp với các phép toán trong logic hình thành nên công thức trong logic mệnh đề. Bài toán chứng minh các công thức logic đồng nhất đúng, đồng nhất sai sẽ có ứng dụng rất lớn trong việc chứng minh một chương trình máy tính chạy đúng hay sai.

2.3.3 Bài toán kiểm tra tính rỗng

Kỹ thuật kiểm tra tính rỗng của ngôn ngữ là kỹ thuật chính trong các công cụ kiểm chứng phần mềm hiện này. Tư tưởng chính của kỹ thuật này là xây dựng ngôn ngữ phần bù của tính chất cần kiểm tra sau đó lấy tích đồng bộ với ngôn ngữ của mô hình cần kiểm tra. Nếu tích đồng bộ rỗng thì có thể khẳng định mô hình hệ thống thỏa tính chất cần kiểm tra. Phổ biến, người ta dùng ôtômát thời gian là mô hình của hệ thống và sử dụng các loại logic để đặc tả tính chất của hệ thống.

2.4 Công cụ kiểm chứng hệ thống thời gian thực

Hiện nay có nhiều công cụ cho kiểm chứng hệ thống thời gian thực như UPPAAL, Kronos, HyTech, v.v. Các công cụ này đều áp dụng ô tô mát thời gian hoặc ôtômát lai (Hybrid automata).

2.5 Tổng kết chương

Chương này đã trình bày các kiến thức cơ bản, và được sắp xếp một cách hệ thống làm cơ sở cho các nghiên cứu trong các phần tiếp theo của luận án.

Chương 3

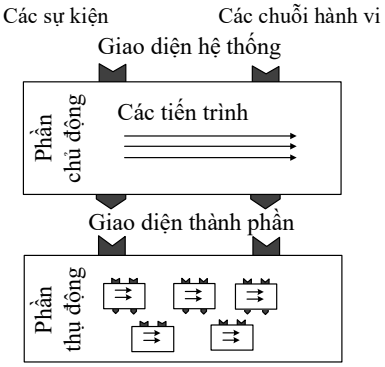
Mô hình thành phần phần mềm thời gian thực và thể thức tương tác tương tranh

3.1 Giới thiệu

Luận án mở rộng mô hình PECOS bằng cách bổ sung thể thức tương tác tương tranh thời gian thực với khả năng đặc tả cả thành phần chức năng và phi chức năng. Dựa trên cách tiếp cận *Tính đúng đắn dựa trên cách xây dựng* và *Thiết kế bằng hợp đồng* được sử dụng để đặc tả thành phần phần mềm thời gian thực bằng các hợp đồng thời gian.

3.2 Mô hình thành phần phần mềm thời gian thực

Trong phần này, luận án trình bày kiến trúc cũng như các thành phần cơ bản trong mô hình mà luận án muốn phát triển được minh họa trong hình 3.1.



Hình 3.1: Minh họa mô hình phần mềm

3.3 Thể thức tương tác tương tranh ràng buộc thời gian

3.3.1 Thể thức tương tác tương tranh

Thể thức tương tác là một tập các quy tắc quy định cách thức các dịch vụ (phương thức) trong thành phần phần mềm được thực thi như thế nào gồm thứ tự lời gọi và các ràng buộc kèm theo như ràng buộc thời gian, tài nguyên hệ thống. Và cách thức các dịch vụ thực thi song song, hay nối tiếp. Do đó, thể thức tương tác được định nghĩa như sau:

Định nghĩa 3.1 (Thể thức tương tác)**.** *Một thể thức tương tác π là một bộ $\langle(\Sigma_1, \mathbb{E}_1), \ldots, (\Sigma_k, \mathbb{E}_k), \delta\rangle$. Trong đó $\delta : \Omega \to \mathbb{R}_{\geq 0}$, và với mọi $i = \overline{1,k}$, $\mathbb{E}_i$ là biểu thức chính quy trên bảng chữ cái Σ_i.*

Định nghĩa 3.2 (Sự tuân thủ thể thức)**.** *Một từ thời gian $\omega = (a_1, \tau_1)(a_2, \tau_2)\ldots(a_n, \tau_n)$ tuân thủ thể thức π, ký hiệu $\omega \models \pi$ khi và chỉ khi với mọi $i \leq k$ thì*

(i) $untimed(\omega)|_{\Sigma_i} \in \mathcal{L}(\mathbb{E}_i)$, *và*
(ii) đặt $untimed(\omega) = a_{j_1}a_{j_2}\ldots a_{j_{m_i}}$, *thì* $\tau_{j_{l+1}} - \tau_{j_l} \geq \delta(a_{j_l})$ *với mọi* $l < m_i$

3.3.2 Phép chiếu

Giả sử cho một chuỗi $\zeta = abcdeabd$ và một tập hợp các phần tử $\Sigma = \{a, b, c, d\}$, $\zeta|_\Sigma$ là một xâu nhận được từ xâu ζ bằng cách loại đi những phần tử không thuộc Σ. Ta có $\zeta|_\Sigma = abcdabd$.

3.3.3 Thuật toán kiểm chứng tính cắm được

Môi trường muốn sử dụng các dịch vụ từ các thành phần phần mềm thì môi trường phải tuân thủ các thể thức của thành phần phần mềm.

Cho một ôtômát thời gian $M = \langle L, \ell_0, \Sigma, \complement, T, \Gamma$. Đặt $\mathcal{L}(M)$ là ngôn ngữ của môi trường, $\mathcal{L}(\mathcal{P}(M))$ là ngôn ngữ của ôtômát vùng của M. Ta có kết quả sau:

Định lý 3.1.

(i) Đối với ôtômát thời gian M, $untimed(\mathcal{L}(M)) = \mathcal{L}(\mathcal{P}(M))$. *Do đó, bài toán quyết định được đối với ôtômát* M *là quyết định được.*
(ii) Nếu $\langle s_0, \nu_0\rangle \xrightarrow[\tau_1]{e_1} \langle s_1, \nu_1\rangle \xrightarrow[\tau_2]{e_2} \ldots \xrightarrow[\tau_m]{e_m} \langle s_m, \nu_m\rangle$ *là một dãy thực thi từ trạng thái ban đầu của* M *thì* $\langle s_0, [\nu_0]\rangle \longrightarrow^{e_1} \langle s_1, [\nu_1]\rangle \longrightarrow^{e_2} \ldots \longrightarrow^{e_m} \langle s_m, [\nu_m]\rangle$ *là dãy thực thi của* $\mathcal{P}(M)$, *và trái lại, nếu* $\langle s_0, [\nu_0]\rangle \longrightarrow^{e_1} \langle s_1, [\nu_1]\rangle \longrightarrow^{e_2} \ldots \longrightarrow^{e_m} \langle s_m, [\nu_m]\rangle$ *là một dãy thực thi trong* $\mathcal{P}(M)$ *thì tồn tại* $\tau_1, \ldots, \tau_m$ *sao cho* $\langle s_0, \nu_0\rangle \xrightarrow[\tau_1]{e_1} \langle s_1, \nu_1\rangle \xrightarrow[\tau_2]{e_2} \ldots \xrightarrow[\tau_m]{e_m} \langle s_m, \nu_m\rangle$ *là một dãy thực thi từ trạng thái ban đầu của* M.

Đối với ôtômát M, kích thước của M (số các dịch chuyển và vị trí) được ký hiệu bởi $|M|$. Bây giờ chúng ta quay trở lại bài toán quyết định được, nếu $untimed(\mathcal{L}(\mathcal{A}))|_{\Sigma_i} \subseteq \mathcal{L}(\mathbb{E}_i)$ đối với ôtômát thời gian $\mathcal{A}$ đã cho. Điều này chỉ ra rằng bài toán quyết định được là giải quyết được, và chỉ là một hệ quả của Định lý 3.1.

Định lý 3.2. *Cho biểu thức chính quy* $\mathbb{E}_i$ *và ôtômát thời gian* $\mathcal{A}$, *bài toán* $untimed(\mathcal{L}(\mathcal{A}))|_{\Sigma_i} \subseteq \mathcal{L}(\mathbb{E}_i)$ *là quyết định được trong thời gian* $\mathcal{O}(|\mathcal{P}(\mathcal{A})|.|\mathcal{L}(\mathbb{E}_i)|)$.

Định lý 3.3. *Bài toán "liệu một ôtômát thời gian* $\mathcal{A}$ *đã cho tuân thủ thể thức tương tác tương tranh thời gian thực* π*" là quyết định được trong thời gian* $\mathcal{O}(|\mathcal{P}(\mathcal{A}')|^2)$.

3.4 Thể thức tương tác thời gian thực ràng buộc tài nguyên

3.4.1 Thể thức thời gian tài nguyên

Mỗi dịch vụ trong Ω cần một khoảng thời gian để thực thi, tiêu thụ và sử dụng một lượng tài nguyên. Đặt $\mathbb{R}_{\geq 0}$ là tập số thực không âm biểu diễn miền thời gian. Đặt RES là tập các véc tơ, $RES = \{res_1, res_2, \ldots, res_h\}$, mỗi véc

tơ res_i, $i = \overline{1,h}$, có n phần tử, mỗi phần tử là một số nguyên đại diện cho đại lượng tài nguyên mà nó biểu diễn. Mỗi véc tơ res_i liên quan đến dịch vụ thứ i trong một thành phần phần mềm. Các yếu tố thời gian và yếu tố tài nguyên được đặc tả bằng một cặp ánh xạ $\mathfrak{d} \widehat{=} (\mathfrak{d}_t, \mathfrak{d}_R)$, trong đó $\mathfrak{d}_t : \Omega \to \mathbb{R}_{\geq 0}$, $\mathfrak{d}_R : \Omega \to RES$. Do đó, thể thức được định nghĩa như sau:

Định nghĩa 3.3 (Thể thức tương tác thời gian tài nguyên)**.** *Một thể thức tương tác $\wp$ là một bộ $\langle (\Sigma_1, \mathbb{E}_1), \ldots, (\Sigma_k, \mathbb{E}_k), \mathfrak{d} \rangle$. Trong đó $\mathfrak{d} \widehat{=} (\mathfrak{d}_t, \mathfrak{d}_R)$, $\mathfrak{d}_t : \Omega \to \mathbb{R}_{\geq 0}$, $\mathfrak{d}_R : \Omega \to RES$, và với mọi $i = \overline{1,k}$, $\mathbb{E}_i$ là biểu thức chính quy trên bảng chữ cái Σ_i.*

Các ràng buộc tài nguyên bị tăng hoặc giảm trong quá trình hệ thống hoạt động. Luận án cần bổ sung các phép toán và việc tính toán sẽ tùy thuộc vào loại tài nguyên là *Tiêu thụ* hay là *Chiếm dụng* và phép ghép song song hoặc nối tiếp của các thành phần phần mềm. Phép toán $\oplus$ và $\ominus$ được minh họa trong bảng sau.

Bảng 3.1: Liệt kê các toán tử tăng giảm và ước lượng thành phần tài nguyên

Toán tử	Loại tài nguyên	Tiêu chuẩn	Ghép song song	Ghép nối tiếp
$\oplus$	Chiếm dụng	$R_{1\|_u} + R_{2\|_u}$	$+R_{1\|_u} + R_{2\|_u}$	$+Max(R_{1\|_u}, R_{2\|_u})$
	Tiêu thụ	$R_{1\|_c} + R_{2\|_c}$	$+R_{1\|_c} + R_{2\|_c}$	$+R_{1\|_c} + R_{2\|_c}$
$\ominus$	Chiếm dụng	$R_{1\|_u} - R_{2\|_u}$	$-R_{1\|_u} - R_{2\|_u}$	$-Max(R_{1\|_u}, R_{2\|_u})$
	Tiêu thụ	$R_{1\|_c} - R_{2\|_c}$	$-R_{1\|_c} - R_{2\|_c}$	$-R_{1\|_c} - R_{2\|_c}$

Định nghĩa 3.4 (Sự tuân thủ thể thức thời gian tài nguyên)**.** *Một từ trọng số $\omega = (a_1, t_1, r_1)(a_2, t_2, r_2) \ldots (a_n, t_n, r_n)$ tuân thủ thể thức $\wp$, ký hiệu $\omega \models \wp$ khi và chỉ khi với mọi $i \leq k$ thì*

(i) $unpricedtimed(\omega)|_{\Sigma_i} \in \mathcal{L}(\mathbb{E}_i)$, và

(ii) đặt $unpricedtimed(\omega)|_{\Sigma_i} = a_{j_1} a_{j_2} \ldots a_{j_{m_i}}$, thì $t_{j_{l+1}} - t_{j_l} \geq \mathfrak{d}_t(a_{j_l})$, thì $r_{j_{l+1}} \ominus r_{j_l} \geq \mathfrak{d}_R(a_{j_l})$, với mọi $l < m_i$.

3.4.2 Mô hình hóa và sự tuân thủ thể thức thời gian tài nguyên

3.4.2.1 Ôtô mát trọng số ràng buộc tài nguyên

Phần này luận án trình bày ôtômát trọng số ràng buộc tài nguyên, được sử dụng để mô hình hóa các chuỗi hành vi của môi trường với ràng buộc tài nguyên. Định nghĩa ôtômát trọng số ràng buộc tài nguyên được khái quát như sau:

Định nghĩa 3.5 (Ôtômát trọng số ràng buộc tài nguyên)**.** *Một ôtômát trọng số trên tập hữu hạn đồng hồ $\complement$ và tập hữu hạn các thành phần tài nguyên R là một bộ $\mathcal{M} = (L, \ell_0, \Sigma, \complement, T, \lambda, \Gamma)$, trong đó L là tập các vị trí, ℓ_0 là vị trí khởi tạo, $T \subseteq L \times \Sigma \times \Phi(\complement) \times 2^{\complement} \times L$ là tập các dịch chuyển, Γ là tập vị trí chấp nhận được, $\lambda : L \cup T \to R \cup RES$ gán mỗi cạnh một bộ giá trị của thành phần tài nguyên RES tương ứng với hành động trong bảng chữ cái Σ, và gán mỗi vị trí một bộ giá trị R.*

Một trạng thái của ôtômát trọng số ràng buộc tài nguyên là một bộ $\langle \ell, \nu \rangle$, trong đó $\ell \in L$, $\nu \in \mathbb{R}^{\complement}_{\geq 0}$. Ánh xạ λ gán nhãn giá trị tài nguyên $\mathfrak{d}_R(a_i) = res_i$

được sử dụng cho hành động a_i lên cạnh của ôtômát và gán giá trị tài nguyên còn lại $r_i \in R$ của hệ thống cho vị trí ℓ_i của ôtômát.

- Thời gian trôi: $\langle \ell, \nu \rangle \xrightarrow{d} \langle \ell, \nu + d \rangle$ nếu với mọi $d \in \mathbb{R}_{\geq 0}$, $\lambda(\langle \ell, \nu \rangle) = \lambda(\langle \ell, \nu \rangle \xrightarrow{d} \langle \ell, \nu + d \rangle)$. Ánh xạ λ gán $r \ominus_{|_c} (d \otimes \mathfrak{d}_R)$ lên vị trí ℓ, trong đó phép toán $\otimes$, d được nhân lần lượt với các thành phần tiêu thụ trong res, giá trị của các thành phần chiếm dụng được giữ nguyên.
- Dịch chuyển rời rạc: $\langle \ell, v \rangle \xrightarrow{e}, \langle \ell', \nu' \rangle$ nếu tồn tại một dịch chuyển $e = \langle \ell, a, \phi, \hbar, \ell' \rangle \in T$ sao cho $\nu \models \phi$, $\nu' = [\hbar \mapsto 0]\nu$, và $\lambda(\langle \ell', \nu' \rangle) = \lambda(\langle \ell, \nu \rangle \xrightarrow{e} \langle \ell', \nu' \rangle)$, trong đó $r' = r \ominus \mathfrak{d}_R(a)$.

3.4.2.2 Ôtô mát trọng số vùng tài nguyên

Cho một ôtômát trọng số ràng buộc tài nguyên $\mathcal{M}$. Một trạng thái đến được của một ôtômát vùng trọng số tài nguyên là một cặp $\langle \ell, Z \rangle$, trong đó Z là một vùng và ℓ là một vị trí. Trạng thái $\langle \ell, Z \rangle$ biểu diễn tập tất cả các trạng thái $\langle \ell, \nu \rangle$ trong đó $\nu \in Z$. Khi Z là một vùng và $\hbar$ là một tập con đồng hồ, $\hbar \subseteq \complement$. Ký hiệu Z^d là tập các phép gán giá trị đồng hồ trong trường hợp thời gian trôi và $Z^\hbar$ là tập các phép gán trên các đồng hồ được thiết lập lại giá trị. Đó là, $Z^d = \{\nu + d \mid \nu \in Z, d \in \mathbb{R}_{\geq 0}\}$ và $Z^\hbar = \{[\hbar \mapsto 0]\nu \mid \nu \in Z\}$. Cho một trạng thái $\langle \ell, Z \rangle$, đặt Δ_Z là phép định giá đồng hồ duy nhất của vùng Z thỏa $\forall \nu \in Z. \forall x \in \complement. \Delta_Z(x) \leq \nu(x)$.

Định nghĩa 3.6. *Một vùng trọng số ràng buộc tài nguyên $\mathcal{Z}$ là một bộ (Z, μ, η), trong đó Z là một vùng, $\mu \in \mathbb{N}$ miêu tả độ lệch tương đối Δ_Z của vùng Z, và $\eta : \complement \to R$ tính giá trị tài nguyên $\eta(x)$ cho bất kỳ đồng hồ x nào. Đặt $\nu \in \mathcal{Z}$ mỗi khi $\nu \in Z$. Với bất kỳ $\nu \in \mathcal{Z}$ chi phí của phép gán ν trong $\mathcal{Z}$, ký hiệu bởi $\mathsf{Cost}(\nu, \mathcal{Z})$, được tính bằng $\mu \otimes \mathfrak{d}_R \oplus \bigoplus_{x \in \complement} \eta(x) \otimes (\nu(x) - \Delta_Z(x))$.*

3.4.2.3 Sự tuân thủ thể thức thời gian tài nguyên

Định lý 3.4. *Cho một môi trường $\mathcal{E}$ được mô hình hóa bằng ôtômát trọng số rang buộc tài nguyên $\mathcal{A}$ và thành phần phần mềm $\mathbb{C}$ với thể thức $\wp$. $\mathcal{E}$ cắm được vào $\mathbb{C}$ khi và chỉ khi $\mathcal{L}(\mathcal{A}) \models \wp$.*

Độ phức tạp thuật toán về thời gian được tính như sau $\mathcal{O}(|PZ(\mathcal{A}')|.|\mathcal{L}(\mathbb{E})_i|)$.

3.5 Tổng kết chương

Chương 3 đã đề xuất mô hình phần mềm thời gian thực dựa trên thành phần có thể áp dụng cho nhiều hệ thống thời gian thực có các nền tảng phần cứng khác nhau. và đề xuất thể thức tương tác tương tranh được sử dụng để biểu diễn sự tương tác giữa các thành phần phần mềm và giữa các thành phần phần mềm với môi trường trên cả khía cạnh chức năng và phi chức năng.

Chương 4

Phương pháp đặc tả và mô hình hóa giao diện thời gian thực

4.1 Giới thiệu

Một giao diện thời gian thực được coi như một hợp đồng trên tập biến đầu vào và đầu ra X và Y, $X \cap Y = \emptyset$, Đặt ξ là một công thức logic tân từ cấp 1biểu diễn mối quan hệ giữa X và Y, $[b, e]$ là khoảng thời gian tối thiểu và tối đa để một thành phần phần mềm cung cấp dịch vụ. Đặt $\mathcal{F}(X \cup Y)$ là tập của tất cả các công thức logic trên $(X \cup Y)$. Đặt $Time$ là tập thời gian, và $Intv$ là tập tất cả các khoảng thời gian trong $Time$. Đặt $\mathcal{V}(X \cup Y)$ biểu thị tập tất cả phép gán các giá trị trên các đầu vào, đầu ra $(X \cup Y)$, nghĩa là với a bất kỳ, $a \in \mathcal{V}(X \cup Y)$ thì ánh xạ $a : (X \cup Y) \to \mathbb{R}_{\geq 0}$. Một trạng thái là một cặp (a, τ), trong đó a là phép gán giá trị, τ là một thời điểm. Đặt $S = \mathcal{V}(X \cup Y) \times Time$ là tập tất cả các trạng thái.

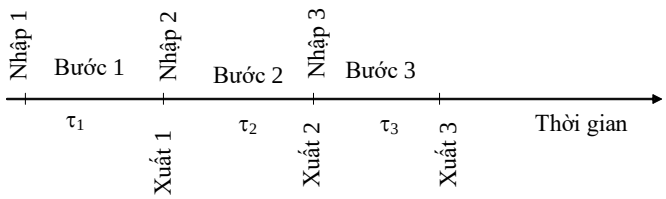
Định nghĩa 4.1. *Một giao diện thời gian thực là một bộ $\langle X, Y, \xi \rangle$, trong đó X và Y tương ứng là các tập hữu hạn đầu vào và đầu ra của giao diện, $\xi : S^* \to \mathcal{F}(X \cup Y) \times Intv$, ở đây $S = \mathcal{V}(X \cup Y) \times Time$.*

Ví dụ 4.1. *Ví dụ này minh họa một giao diện thành phần với tập biến đầu vào chỉ có một biến $\{x\}$, tập đầu ra có một biến $\{y\}$.*

$$\mathcal{I} = \langle \{x\}, \{y\}, (x > 0 \vdash x > 0 \wedge y = x + 1), [1, 3] \rangle$$

Định nghĩa 4.2. *Một dãy thực thi của giao diện $\mathbb{I} = \langle X, Y, \xi \rangle$ là một dãy các trạng thái $s_1 s_2 \dots s_n$, $i = \overline{1, n}$, ở đây $s_i = (a_i, \tau_i)$, sao cho $a_i \models \Phi_i$ và $\tau_i \in \mathsf{I}_i$. Công thức $\xi(s_1 \dots s_{i-1}) = (\Phi_i, \mathsf{I}_i)$ với mọi $i = \overline{1, n}$. Tập tất các các dãy thực thi của $\mathbb{I}$ được ký hiệu bằng $\mathsf{S}(\mathbb{I})$. Một dãy thực thi cũng được gọi là một dãy trạng thái đến được.*

Hình 4.1 minh họa sự một dãy thực thi của một giao diện.



Hình 4.1: Minh họa sự thực thi theo thời gian của giao diện

Định nghĩa 4.3. *Một môi trường là một bộ các thành phần $\langle X, Y, h_X, h_Y \rangle$, ký hiệu là E, ở đây X và Y như trong Định nghĩa 4.1, và $h_X : (\mathcal{V}(X \cup Y) \times$*

Time)$^* \to \mathcal{F}(X)$ *and* $h_Y : (\mathcal{V}(X \cup Y) \times Time)^* \to \mathcal{F}(X \cup Y) \times Intv$. *Tập tất cả các dãy thực thi của* E *được ký hiệu là* $\mathsf{S}(E)$.

Ánh xạ h_X đảm bảo rằng tập đầu vào X mà môi trường cung cấp cho tại một trạng thái đã cho, trong khi h_Y thể hiện điều mong muốn của môi trường trên các tập đầu ra Y và thời gian để nhận giá trị tại đầu ra. Môi trường E hoạt động được nếu h_X và h_Y thỏa tất cả các dãy trạng thái đến được

Định nghĩa 4.4 (Phép cắm)**.** *Một môi trường* $E = \langle X, Y, h_X, h_Y \rangle$ *có thể cắm được vào giao diện* $\mathbb{I} = \langle X', Y', \xi \rangle$ *khi và chỉ khi* $X' = X$, $Y' = Y$ *và*

1. $h_X(\epsilon) \Rightarrow in(\xi_f(\epsilon))$, $\xi_f(\epsilon) \wedge h_X(\epsilon) \Rightarrow h_{Y_f}(\epsilon)$, *và* $\xi_\tau(\epsilon) \subseteq h_{Y_\tau}(\epsilon)$. *Trạng thái* ϵ *gọi là trạng thái đến được.*
2. *Với tất cả các trạng thái đến được* $s \in \mathcal{S}^*$ *sao cho* $h_X(s)$ *thỏa được,* $h_X(s) \Rightarrow in(\xi_f(s))$, $\xi_f(s) \wedge h_X(s) \Rightarrow h_{Y_f}(s)$, *và* $\xi_\tau(s) \subseteq h_{Y_\tau}(s)$ *đúng. Đối với* $(a, \tau) \in \mathcal{S}$ *sao cho* $a \models \xi_f(s) \wedge h_X(s)$ *và* $\tau \in h_{Y_\tau}(s)$ *dãy trạng thái* $s.(a, \tau)$ *cũng được gọi là đến được.*

Luận án ký hiệu tập tất cả các dãy trạng thái đến được của môi trường E *cắm vào giao diện* $\mathbb{I}$ *là* $\mathsf{S}(E, \mathbb{I})$.

Định nghĩa 4.5 (Tương đương môi trường)**.** *Hai giao diện* $\mathbb{I}$ *and* $\mathbb{I}'$ *tương đương nhau tương ứng với môi trường* E, *ký hiệu* $\mathbb{I} \equiv_E \mathbb{I}'$, *khi và chỉ khi* E *cắm được vào* $\mathbb{I}$ *và* E *cũng cắm được vào* $\mathbb{I}'$. *Hai giao diện* $\mathbb{I}$ *và* $\mathbb{I}'$ *tương đương về mặt môi trường khi và chỉ khi* $\mathbb{I} \equiv_E \mathbb{I}'$ *đối với mọi môi trường* E.

Định lý 4.1 (Định lý tương đương môi trường)**.** $\mathbb{I} \equiv_E \mathbb{I}'$ *đối với mọi môi trường* E *khi và chỉ khi* $\mathbb{I} \equiv \mathbb{I}'$.

4.2 Ghép giao diện thành phần

Phép ghép các giao diện thành phần là một trong những nội dung quan trọng trong lý thuyết giao diện. Luận án xét hai trường hợp của phép ghép: Ghép song song và ghép nối tiếp. và chỉ ra một số kết quả đạt được

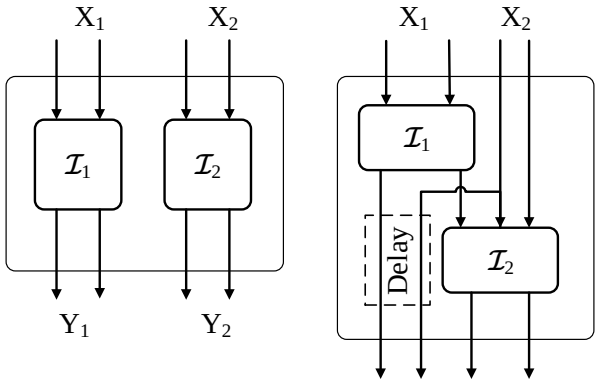
Định nghĩa 4.6 (Ghép song song giao diện)**.**

Định lý 4.2. *Đặt* $\mathbb{I}'' = \mathbb{I}||\mathbb{I}'$. *Một dãy trạng thái* $(a_1, \tau_1) \dots (a_n, \tau_n)$ *là một dãy trạng thái đến được của* $\mathbb{I}''$ *khi và chỉ khi* $(a_1|_{X \cup Y}, \tau_1) \dots (a_n|_{X \cup Y}, \tau_n)$ *là dãy trạng thái đến được của* $\mathbb{I}$ *và* $(a_1|_{X' \cup Y'}, \tau_1) \dots (a_n|_{X' \cup Y'}, \tau_n)$ *là dãy trạng thái đến được của* $\mathbb{I}'$.

Ngoài cách ghép song song hai giao diện, còn một cách ghép nối tiếp hai giao diện với nhau

Định nghĩa 4.7 (Ghép nối tiếp giao diện)**.**

Phép ghép song song và nối tiếp của hai giao diện $\mathbb{I}$ và $\mathbb{I}'$ được minh họa trong Hình 4.2.



Hình 4.2: Minh họa phép song song (a) phép nối tiếp (b)

Định lý 4.3. *Đặt* $\mathbb{I}$, $\mathbb{I}'$ *và* $\mathbb{I}''$ *là các giao diện,* θ *và* θ' *là các phép ghép tương ứng giữa* $\mathbb{I}$ *với* $\mathbb{I}'$ *và giữa* $\mathbb{I}'$ *với* $\mathbb{I}''$*. Suy ra những điều sau đây đúng:*

(i) $\mathbb{I}||\mathbb{I}' \equiv \mathbb{I}'||\mathbb{I}$,
(ii) $(\mathbb{I}||\mathbb{I}')||\mathbb{I}'' \equiv \mathbb{I}||(\mathbb{I}'||\mathbb{I}'')$,
(iii) $(\mathbb{I}._\theta\mathbb{I}').._{\theta'}\mathbb{I}'' \equiv \mathbb{I}.._\theta(\mathbb{I}'.._{\theta'}\mathbb{I}'')$.

4.3 Sự làm mịn giao diện thành phần

Sự làm mịn giao diện có nghĩa là một giao diện có thể được thay bằng một giao diện có chất lượng tốt hơn. Dựa trên logic Hoare, định nghĩa về làm mịn giao diện như sau:

Định nghĩa 4.8. *Đặt* $\mathbb{I} = \langle X, Y, \xi\rangle$ *và* $\mathbb{I}' = \langle X', Y', \xi'\rangle$ *là hai giao diện.* $\mathbb{I}$ *được làm mịn bởi* $\mathbb{I}'$ *(hoặc* $\mathbb{I}'$ *làm mịn* $\mathbb{I}$*), ký hiệu* $\mathbb{I} \sqsubseteq \mathbb{I}'$*, khi và chỉ khi* $X = X'$*,* $Y = Y'$ *và với mọi* $s \in \mathsf{S}(\mathbb{I}) \cap \mathsf{S}(\mathbb{I}')$ *những điều sau đây đúng:* $in(\xi_f(s)) \Rightarrow in(\xi'_f(s))$*,* $in(\xi_f(s)) \wedge \xi'_f(s) \Rightarrow \xi_f(s)$*, và* $\xi'_\tau(s) \subseteq \xi_\tau(s)$*.*

Do vậy, $\mathbb{I}'$ cung cấp các dịch vụ tốt hơn theo nghĩa là nó cung cấp cùng các dịch vụ cho môi trường với các điều kiện lỏng hơn của môi trường trong hợp đồng của giao diện. Định lý sau kiểm chứng định nghĩa của sự làm mịn giao diện.

Định lý 4.4. *Đặt* $\mathbb{I} \sqsubseteq \mathbb{I}'$*, và môi trường* E *cắm vào* $\mathbb{I}'$*. Thì môi trường* E *cũng cắm vào* $\mathbb{I}$ *và* $\mathsf{S}(E, \mathbb{I}') \subseteq \mathsf{S}(E, \mathbb{I})$ *đúng.*

Các phép ghép giao diện đã định nghĩa ở Phần 4.2 bảo toàn sự làm mịn như được phát biểu dưới đây.

Định lý 4.5 (Bảo toàn sự làm mịn)**.** *Đặt* $\mathbb{I}, \mathbb{I}'$ *và* $\mathbb{I}''$ *là các giao diện sao cho* $\mathbb{I} \sqsubseteq \mathbb{I}'$*, và các phép ghép* $||$ *và* θ *là phép ghép nối tiếp giữa* $\mathbb{I}$ *và* $\mathbb{I}''$*. Giả sử rằng các điều kiện cho các phép ghép là thỏa được thì* $(\mathbb{I}||\mathbb{I}'') \sqsubseteq (\mathbb{I}'||\mathbb{I}'')$ *và* $(\mathbb{I}._\theta\mathbb{I}'') \sqsubseteq (\mathbb{I}'.._\theta\mathbb{I}'')$*.*

4.4 Mô hình hóa hành vi của giao diện

Do hành vi của giao diện thành phần thời gian thực là vô hạn, chúng ta cần mô hình hóa các hành vi này để có thể kiểm soát được chất lượng của các dịch vụ của thành phần phần mềm do môi trường sử dụng.

Định nghĩa 4.9 (Ôtômát khoảng)**.** *Một ôtômát khoảng trên tập đầu vào X, đầu ra Y là một bộ $M_{\mathbb{I}} = \langle Q, \Sigma, X, Y, q_0, T, \Lambda, \Gamma \rangle$, trong đó $X \cap Y = \emptyset$, gồm*

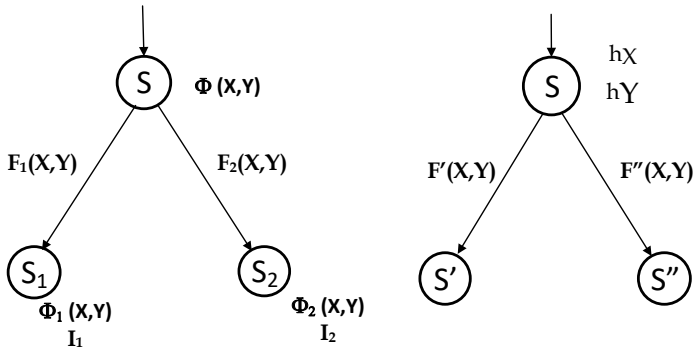
- *Q là tập hữu hạn các trạng thái,*
- *Σ là tập hữu hạn các hành động,*
- *$q_0 \in Q$ là trạng thái khởi tạo của M,*
- *$T \subseteq S \times \Sigma \times S$ là tập các quan hệ dịch chuyển thời gian,*
- *$\Lambda = (\Lambda_f, \Lambda_\tau)$, $\Lambda_f : S \to \mathcal{F}(X \cup Y)$ là hàm gán nhãn trên mỗi trạng thái của M với một biểu thức logic tân từ cấp 1 chỉ mối quan hệ giữa đầu vào X với đầu ra Y, $\Lambda_\tau : S \to Intv$ gán nhãn thời gian lên trạng thái của M.*
- *$\Gamma \subseteq Q$ là tập trạng thái kết thúc.*

Để một ôtômát khoảng giao diện biểu diễn được hành vi của giao diện thành phần, luận án định nghĩa giao diện cho ôtômát khoảng như sau:

Định nghĩa 4.10 (Ôtômát khoảng giao diện)**.** *Một ôtômát khoảng giao diện $\mathbb{I}(M_{\mathbb{I}})$ là bộ ba $\langle X \cup \{\ddot{x}\}, Y \cup \{\ddot{y}\}, \xi \rangle$ trong đó X và Y là tập hữu hạn tương ứng với đầu vào và đâu ra. $\ddot{x}$ là biến mới trên tập hành động Σ, và $\ddot{y}$ là biến mới trên tập trạng thái Q. Quan hệ $\xi(\sigma.(a,\tau)) = (\Lambda_f(s) \wedge \vee_{(s,c,s') \in T}(\ddot{x} = c) \Rightarrow (\ddot{y} = s'), \Lambda_\tau(s))$, $\tau \in \Lambda_\tau(s)$, trong đó $a(\ddot{y}) = s'$ và σ là dãy tính toán dẫn đến s.*

Định nghĩa 4.11. *Một dãy thực thi của một ôtômát khoảng giao diện $\mathbb{I}(M) = \langle X \cup \{\ddot{x}\}, Y \cup \{\ddot{y}\}, \xi \rangle$ là một dãy các trạng thái $s_1 s_2 \dots s_n$ sao cho $a_i \models \Lambda_f(s_i)$ và $\tau_i \in \Lambda_\tau(s_i)$, ở đây $s_i = (a_i, \tau_i)$ và $\xi(s_1 \dots s_{i-1}) = \Lambda_i$, $i = \overline{1,n}$. Tập tất cả các trạng thái của $\mathbb{I}(M)$ ký hiệu bởi $\mathsf{S}(\mathbb{I}(M))$.*

Định nghĩa 4.12 (Ôtômát khoảng cho môi trường)**.** *Một ôtômát khoảng hữu hạn môi trường M_E được định nghĩa giống như định nghĩa 4.9 ngoại trừ hàm gán nhãn $\Lambda_f : S \to \mathcal{F}(X) \times \mathcal{F}(X \cup Y)$ và $\Lambda_\tau : S \to Intv$.*



Hình 4.3: Minh họa ôtômát khoảng giao diện và môi trường

Tương tự như trường hợp của ôtômát khoảng giao diện, một ôtômát khoảng môi trường được định nghĩa như dưới đây:

Định nghĩa 4.13. *Ôtômát khoảng môi trường là một bộ $\mathcal{E}(M_E) = \langle X \cup \{\ddot{x}\}, Y \cup \{\ddot{y}\}, h_X \wedge in \in \ddot{y}^{in}, h_Y \rangle$ trong đó $h_X(\sigma.(a,\tau)) = \Lambda_X(s) \wedge i \in s^+$ và $h_Y(\sigma.(a,\tau)) = (\Lambda_Y(s) \wedge (s, \ddot{x}, \ddot{y}) \in T, \Lambda_{Y_\tau})$.*

Định lý 4.6. *Cho ôtômát khoảng $M_{\mathbb{I}}$ và một ôtômát khoảng cho môi trường M_E. Giao diện môi trường $\mathcal{E}(M_E)$ cắm vào giao diện $\mathbb{I}(M_{\mathbb{I}})$ khi và chỉ khi M_E là đồ thị con của $M_{\mathbb{I}}$, và với mọi $s \in \mathsf{S}(E)$, $\Lambda_X(s) \Rightarrow in(\Lambda_f(s)), \Lambda_X(s) \wedge \Lambda_f(s) \Rightarrow \Lambda_Y(s)$ và $\Lambda_{Y_\tau}(s) \subseteq \Lambda_\tau(s)$.*

4.5 Tổng kết chương

Chương này đã đạt được một số kết quả: Đặc tả thành phần phần mềm thời gian thực bằng giao diện thành phần thời gian thực, mô hình hóa các hành vi của giao diện thành phần và môi trường, và kiểm chứng cắm được của môi trường vào giao diện thành phần.

Chương 5

Phương pháp đặc tả và kiểm chứng bằng hợp đồng thời gian thực

5.1 Giới thiệu

Trong chương này, luận án đề xuất một cải tiến kỹ thuật đặc tả thành phần phần mềm thời gian thực bằng hợp đồng thời gian. Kỹ thuật này hình hướng đến mục tiêu giảm thiểu lỗi trong quá trình phát triển hệ thống thời gian thực dựa trên thành phần.

5.2 Thiết kế thời gian

Theo lý thuyết phát triển hệ thống thời gian thực dựa trên thành phần, luận án đặc tả các phương thức trong thành phần phần mềm bằng thiết kế thời gian, và được định nghĩa như sau:

Định nghĩa 5.1 (Thiết kế thời gian)**.** *Một* thiết kế thời gian *trên tập các biến đầu vào, đầu ra X, Y là một bộ các thành phần $D \triangleq \langle V, \mathcal{R}, d\rangle$, trong đó $V = X \cup Y$, $\mathcal{R}$ là một công thức logic tân từ cấp 1 có dạng $\alpha \vdash \beta$ biểu diễn quan hệ giữa các biến đầu vào và các biến đầu ra, ở đây α là tiền điều kiện trên tập các biến $A \subseteq X$, β là hậu điều kiện trên các tập biến đầu ra $B \subseteq Y$, d là số nguyên dương biểu diễn khoảng thời gian tối thiểu thực thi một dịch vụ.*

Giao diện của thành phần được định nghĩa như sau:

Định nghĩa 5.2 (Giao diện thành phần)**.** *Giao diện thành phần là một cặp $\mathcal{I} \triangleq (\mathcal{I}_p, \mathcal{I}_r)$, trong đó $\mathcal{I}_p \triangleq \langle \mathcal{F}d_p, Md_p\rangle$ gọi là giao diện cung cấp của $\mathcal{I}$, và $\mathcal{I}_r \triangleq \langle \mathcal{F}d_r, Md_r\rangle$ gọi là giao diện yêu cầu của $\mathcal{I}$.*

5.3 Hợp đồng thời gian

Tiếp theo, luận án định nghĩa *hợp đồng* cho thành phần phần mềm, hợp đồng sẽ cho biết cách thức một thành phần phần mềm được sử dụng như thế nào.

Định nghĩa 5.3 (Hợp đồng). *Một hợp đồng là một bộ* $\mathfrak{C} = \langle \mathcal{I}, I, M_{Spec}, \mathcal{I}nv, \pi \rangle$, *trong đó*

- $\mathcal{I} = (\mathcal{I}_p, \mathcal{I}_r)$ *là một giao diện. Đặt* $Md = Md_p \cup Md_r$, $\mathcal{F}d = \mathcal{F}d_p \cup \mathcal{F}d_r$.
- I *là sự khởi tạo các giá trị ban đầu cho từng thuộc tính trong tập* $\mathcal{F}d$.
- M_{Spec} *là đặc tả phương thức, chúng liên quan đến từng phương thức* $op(X, Y)$ *trong tập* $Md = Md_p \cup Md_r$ *tương ứng với từng* thiết kế $D = \langle V, \mathcal{R}, d \rangle$.
- $\mathcal{I}nv$ *là tập các* ràng buộc bất biến của hợp đồng *được biểu diễn bằng cặp* $(\mathcal{I}nv_p, \mathcal{I}nv_r)$, *trong đó* $\mathcal{I}nv_p$ *và* $\mathcal{I}nv_r$ *là công thức logic LTL tương ứng ràng buộc trên tập thuộc tính cung cấp và thuộc tính yêu cầu.*
- π *là một thể thức tương tác tương tranh thời gian thực.*

5.4 Ghép hợp đồng thời gian

Để phát triển những hệ thống phức tạp, chúng ta ghép các thành phần phần mềm với nhau cho đến khi thỏa các yêu cầu hệ thống bằng cách ghép các hợp đồng theo cách ghép song song, nối tiếp hoặc phép cắm. Sau đây luận án định nghĩa phép ghép song song hai hợp đồng.

Định nghĩa 5.4 (Ghép song song hai hợp đồng thời gian). *Cho hai hợp đồng có khả năng ghép được* $\mathfrak{C}_1 = \langle \mathcal{I}_1, I_1, M_{Spec_1}, \mathcal{I}nv_1, \pi_1 \rangle$ *và hợp đồng* $\mathfrak{C}_2 = \langle \mathcal{I}_2, I_2, M_{Spec_2}, \mathcal{I}nv_2, \pi_2 \rangle$. *Phép ghép song song của hai hợp đồng* $\mathfrak{C}_1$ *và* $\mathfrak{C}_2$ *ký hiệu là* $\mathfrak{C}_{||} = \mathfrak{C}_1 || \mathfrak{C}_2$ *là một hợp đồng* $\mathfrak{C}_{||} = \langle \mathcal{I}, I, M_{Spec}, \mathcal{I}nv, \pi \rangle$, *các thành phần trong hợp đồng mới được tính như sau:*

- $\mathcal{I} = (\mathcal{I}_{p1} \cup \mathcal{I}_{p2}, \mathcal{I}_{r1} \cup \mathcal{I}_{r2})$
- $I = I_1 \cup I_2$
- $M_{Spec} = M_{Spec_1} \cup M_{Spec_2}$, *các thành phần trong* Md *được tính như sau:* $Md_r = Md_{r1} \cup Md_{r2} \setminus Shared(\mathfrak{C}_1, \mathfrak{C}_2)$, $Md_p = Md_{p1} \cup Md_{p2} \setminus Shared(\mathfrak{C}_1, \mathfrak{C}_2)$.
- Inv *được tính như sau:* $\mathcal{I}nv = \mathcal{I}nv_1 \cup \mathcal{I}nv_2$
- $\pi = \pi_1 || \pi_2$ *là thể thức tương tác tương tranh thời gian thực.*

Ngoài cách ghép song song, còn cách ghép khác đó là phép cắm hai hợp đồng được định nghĩa như sau:

Định nghĩa 5.5 (Cắm hoàn toàn). $\mathfrak{C}_1 \gg \mathfrak{C}_2 = \langle (\mathcal{I}_{p1} \cup \mathcal{I}_{p2}, \mathcal{I}_{r2}), M_{Spec_1}|_{Md_{p_1}} \cup M_{Spec_2}, I_1|_{\mathcal{F}d_{r_1}} \uplus I_2, \mathcal{I}nv, \pi \rangle$, *trong đó* $\mathcal{I}nv = (\mathcal{I}nv_{p1} \wedge \mathcal{I}nv_{p2}), \mathcal{I}nv_{r2}))$ *và* $(I_1 \uplus I_2)(x)$ *được định nghĩa là:*

$$\begin{cases} I_1(x) = I_2(x) & \text{if } x \in dom(I_1) \cap dom(I_2) \\ I_1(x) & \text{if } x \in dom(I_1) \setminus dom(I_2) \\ I_2(x) & \text{if } x \in dom(I_2) \setminus dom(I_1) \end{cases}$$

Đối với thể thức π được tính theo các trường hợp sau:

- *Đầu tiên, hợp đồng $\mathfrak{C}_1 \gg \mathfrak{C}_2$ sẽ cho phép các phương thức trong các thành phần phần mềm độc lập được sử dụng theo cách thức nguyên gốc của chúng. Bởi vậy, $\pi = \pi_1 \cup \pi_2$.*
- *Các phương thức op trong $\mathfrak{C}_2$ mà không bị gọi bởi $\mathfrak{C}_1$, thì π sẽ hoạt động theo cơ chế song song với các phương thức trong $\mathfrak{C}_1$, Do đó $\pi = \pi_1 || (\pi_2 \cap \{op \in Md_{p2} \setminus Md_{r_1}\}^*)$, và*
- *Trong trường hợp còn lại, phương thức op trong $Md_{p2} \cap Md_{r1}$ được sử dụng với một phương thức trong Md_{p1}. Điều này phụ thuộc vào mức độ tương tranh của phương thức op.*

Do đó, π được định nghĩa như sau: $\pi = \pi_1 \cup \pi_2 \cup (\pi_1 || (\pi_2 \cap \{op \in Md_{p2} \setminus Md_{r_1}\}^*)$.

Khi $\mathfrak{C}_1 \gg \mathfrak{C}_2$ được xác định, chúng ta có thể nói $\mathfrak{C}_1$ có thể nối với $\mathfrak{C}_2$. Chú ý rằng khi nối 2 hợp đồng theo cách này, $\mathfrak{C}_1 \gg \mathfrak{C}_2$ không được phép thay thế $\mathfrak{C}_1$ vì nó có thể yêu cầu một số yếu tố từ môi trường còn $\mathfrak{C}_1$ thì không.

5.5 Hệ thống

Hệ thống có hai phần, *phần thụ động* và *phần chủ động*.

Định nghĩa 5.6 (Thành phần phần mềm thụ động)**.** *Thành phần phần mềm thụ động là một bộ $P_{Comp} = \langle \mathfrak{C}, Mcode \rangle$, trong đó:*

- *Hợp đồng $\mathfrak{C} = \langle \mathcal{I}, I, Md_{Spec}, \mathcal{I}nv, \pi \rangle$.*
- *$Mcode$ gán với mỗi phương thức op trong Md_p một thiết kế được xây dựng từ các toán tử cơ sở và các phương thức trong $\mathcal{I}_r$ sao cho với mọi dãy thực thi $w|_{Md_r} \models \pi$. Điều kiện sau sẽ thỏa bởi $Mcode$: $(M_{Spec}(op) \sqsubseteq Mcode(op))$, và $\mathcal{I}nv_p$ được bảo toàn bởi bất kỳ hoạt động nào được sử dụng trong $Mcode$.*
- *Sự thực thi của tất cả các phương thức m sẽ tương thích với cấp độ tương tranh được mô tả trong π, tức là phương thức m hoặc không loại trừ lẫn nhau hoặc có nhiều bản sao của m.*

Hợp đồng $\mathfrak{C}$ được nói là được thực thi bởi P_{Comp}.

Định lý 5.1. *Cho hai hợp đồng $\mathfrak{C}_1$ và $\mathfrak{C}_2$, ta có $\langle \mathfrak{C}_1 \gg \mathfrak{C}_2, Mcode'_1 \cup Mcode_2|_{Md_2 \setminus Md_1} \rangle$ là một thành phần phần mềm.*

Định nghĩa 5.7 (Giao diện của hệ thống)**.** *Giao diện hệ thống là một bộ $\mathcal{I}_S = \langle Evt, \mathcal{F}d, SMd_p \rangle$, trong đó SMd_p là tập hữu hạn các phương thức, $\mathcal{F}d$ là một tập hữu hạn các thuộc tính, Evt là tập hữu hạn các sự kiện.*

Từ giao diện hệ thống, sau đây là định nghĩa hợp đồng hệ thống.

Định nghĩa 5.8 (Hợp đồng của hệ thống)**.** *Hợp đồng hệ thống là một bộ $Sys\mathfrak{C} = \langle \mathcal{I}_s, M_{Specs_S}, \mathcal{I}nv_S, Behav \rangle$, trong đó*

- *$\mathcal{I}_S = \langle Evt, \mathcal{F}d_s, Md_{p_S} \rangle$ là giao diện hệ thống.*

- M_{Specs} *là đặc tả phương thức liên quan đến từng phương thức* $op(X, Y)$ *trong* Md_{ps} *với mỗi thiết kế* $\langle V, \mathcal{R}, d\rangle$*, và*
- $Behav$ *là sự mô tả hành vi bên ngoài là một tập hữu hạn của* $\{e, m | evt \in Evt, m \in Md_{ps}\}^*$*. Mỗi hành vi của* $Behav$ *được gọi là một đặc tả tiến trình.*

Định nghĩa 5.9 (Thành phần phần mềm chủ động)**.** *Thành phần phần mềm chủ động là một bộ* $A_{Comp} = \langle \mathfrak{C}, Sys\mathfrak{C}, Mcode\rangle$ *bao gồm*

- *Một hợp đồng* $\mathfrak{C} = \langle \mathcal{I}, I, M_{Spec}, \mathcal{I}nv, \pi\rangle$ *với giao diện cung cấp rỗng,* $\mathcal{I}_p = (\emptyset, \emptyset)$*.*
- *Một hợp đồng hệ thống* $Sys\mathfrak{C} = \langle I_s, M_{Spec_s}, Inv, Behav\rangle$*.*
- *Một tiến trình thực thi* $Mcode$ *gán mỗi phương thức* op *trong* Md_{ps} *một thiết kế được xây dựng từ các toán tử cơ sở và phương thức trong* $\mathcal{I}_r$ *sao cho với mọi dãy thực thi* $w|_{Md_r} \models \pi$*. Điều kiện sau sẽ thỏa bởi* $Mcode : (M_{Specs}(op) \sqsubseteq Mcode(op))$ *với mọi* $op \in Md_{ps}$*.*

Một hệ thống trong mô hình thành phần của luận án là một *Thành phần phần mềm chủ động* cắm vào một *Thành phần phần mềm thụ động đóng.*

Định nghĩa 5.10 (Hệ thống)**.** *Hệ thống là một cặp thành phần phần mềm chủ động* $A_{Comp} = \langle \mathfrak{C}, Sys\mathfrak{C}, Mcode\rangle$ *và một thành phần phần mềm thụ động* $P_{Comp} = \langle \mathfrak{C}', Mcode'\rangle$*, sao cho* $\mathfrak{C} \gg \mathfrak{C}'$*.*

Định lý 5.2. *Cho* $P'_{Comp} = \langle \mathfrak{C}', Mcode'\rangle$*,* $P''_{Comp} = \langle \mathfrak{C}'', Mcode''\rangle$ *là các thành phần phần mềm thụ động, và thành phần phần mềm chủ động* $A_{Comp} = \langle \mathfrak{C}, Sys\mathfrak{C}, Mcode\rangle$*. Hệ thống* $System \cong \langle A_{Comp}, P'_{Comp}\rangle$ *tương đương với* $System' \cong \langle A_{Comp}, P''_{Comp}\rangle$ *khi và chỉ khi* $\mathfrak{C}' \sqsubseteq \mathfrak{C}''$*.*

5.6 Tổng kết chương

Chương này đã trình bày kỹ thuật đặc tả cũng như phương pháp phát triển phần mềm thời gian thực dựa trên thành phần bằng hợp đồng thời gian thực.

Chương 6

Phương pháp đặc tả và kiểm chứng thành phần phần mềm thời gian thực có ràng buộc tài nguyên

6.1 Giới thiệu

Chương này phát triển mô hình hình thức có khả năng hỗ trợ phân tích, thiết kế hiệu quả các đặc tả yếu tố chức năng và phi chức năng của hệ thống thời gian thực dựa trên thành phần.

6.2 Các nghiên cứu liên quan

6.3 Thiết kế thời gian-tài nguyên

Một phương thức có dạng $op(X, Y)$ có thể được đặc tả bằng một *Thiết kế tài nguyên-thời gian* trên tập các biến đầu vào X và tập các biến đầu ra Y, $X \cap Y = \emptyset$. Đặt $\mathcal{R}$ là công thức logic tân từ cấp 1 biểu diễn mối quan hệ giữa đầu vào và đầu ra.

Định nghĩa 6.1 (Thiết kế thời gian-tài nguyên)**.** *Mỗi phương thức $op(X, Y)$ được đặc tả bởi một thiết kế có dạng $D(op) \cong \langle \vartheta, \xi, \psi, d, \rho \rangle$,*

- *ϑ biểu diễn tập các biến được dùng bởi phương thức op*
- *ξ biểu diễn đặc tả chức năng và là vị từ có dạng như sau:*

$$p \vdash_f \mathcal{R} \;\widehat{=}\; p \Rightarrow \mathcal{R}$$

- *ψ biểu thị đặc tả thành phần phi chức năng có dạng như sau:*

$$q \vdash_n \mathcal{S} \;\widehat{=}\; q \Rightarrow \mathcal{S}$$

- *d là một số nguyên dương, biểu diễn khoảng thời gian tối thiểu mà hệ thống có thể gọi phương thức.*
- *ρ là một véc tơ n phần tử đặc tả lượng tài nguyên sử dụng cho thiết kế.*

Ký hiệu $\vdash_f$ và $\vdash_n$ tương ứng với chức năng và phi chức năng.

Định nghĩa 6.2 (Làm mịn thiết kế)**.** *Cho hai thiết kế của phương thức $op(X, Y)$, $D(op) = (\vartheta, \xi, \psi, d, \rho)$ và $D'(op) = (\vartheta', \xi', \psi', d', \rho')$. D' được nói là làm mịn của D, ký hiệu bởi $D \sqsubseteq D'$ khi và chỉ khi những điều sau thỏa*

- *$p \Rightarrow p'$, $\mathcal{R}' \Rightarrow \mathcal{R}$,*
- *$q \Rightarrow q'$, $\mathcal{S}' \Rightarrow \mathcal{S}$,*
- *$d' \leq d$ và*
- *$\rho' \leq \rho$.*

Định nghĩa 6.3 (Ghép nối tiếp thiết kế)**.** *Đặt* $D_1(op) \mathrel{\widehat{=}} \langle \vartheta_1, \xi_1, \psi_1, d_1, \rho_1 \rangle$ *và* $D_2(op) \mathrel{\widehat{=}} \langle \vartheta_2, \xi_2, \psi_2, d_2, \rho_2 \rangle$ *là hai thiết kế. Một phép nối nối tiếp* $.\theta$ *giữa hai thiết kế được định nghĩa như sau:*

$$D_1.\theta D_2 \mathrel{\widehat{=}} \langle \vartheta, \xi, \psi, d, \rho \rangle, \textit{ trong đó}$$

- $\vartheta \mathrel{\widehat{=}} \vartheta_1 \cup \vartheta_2$
- *Đặt* ξ *kết hợp với bộ tham số* x*, ký hiệu* $\xi(x)$ *thì* $\xi \mathrel{\widehat{=}} \exists v \bullet \xi_1 \wedge \xi_2$.
- $\psi \mathrel{\widehat{=}} \exists \rho_1, \rho_2 \bullet (\psi_1[\rho_1/\rho] \wedge \psi_2[\rho_2/\rho] \wedge \rho|_u = max(\rho_1|_u, \rho_2|_u)) \wedge \rho|_c = \rho_1|_c \oplus \rho_2|_c \wedge d = d_1 + d_2$.

Trong định nghĩa này, biểu thức $\rho|_u$ *có ký hiệu "|" là phép chiếu của* ρ *trên tập các thành phần tài nguyên được* sử dụng *cho thực thi. Biểu thức* $\psi[x_1/x]$*, ký hiệu "/" là phép thế thành phần* x *vào thành phần* x_1 *trong biểu thức* ψ*. Hàm* $max(\rho_1|_u, \rho_2|_u)$ *là hàm tính toán các thành phần tài nguyên bị chiếm dụng và biểu thức* $\rho|_c = \rho_1|_c \oplus \rho_2|_c$ *sẽ tính toán lượng tài nguyên tiêu thụ cho* $D_1.\theta D_2$*. Biểu thức* $d = d_1 + d_2$ *là tính toán khoảng thời gian tĩnh đảm bảo cho thiết kế* $D_1.\theta D_2$ *có thể đáp ứng được yêu cầu môi trường.*

Trong quá trình phát triển hệ thống, có một cách khác để ghép các thiết kế với nhau đó là hai phương thức thực thi tương tranh thì các điều kiện ràng buộc được định nghĩa như sau:

Định nghĩa 6.4 (Ghép tương tranh thiết kế)**.** *Đặt* $D_1 \mathrel{\widehat{=}} \langle \vartheta_1, \xi_1, \psi_1, d_1, \rho_1 \rangle$ *và* $D_2 \mathrel{\widehat{=}} \langle \vartheta_2, \xi_2, \psi_2, d_2, \rho_2 \rangle$ *là hai thiết kế. Một phép đồng bộ* $||$ *giữa hai thiết kế được biểu thị như sau:*

$$D_1||D_2 \mathrel{\widehat{=}} \langle \vartheta, \xi, \psi, d, \rho \rangle, \textit{ trong đó}$$

- $\vartheta \mathrel{\widehat{=}} \vartheta_1 \cup \vartheta_2$
- $\xi \mathrel{\widehat{=}} \xi_1 \wedge \xi_2$.
- *Đặt* $\psi_1 = q_1(\rho_1) \vdash_n \mathcal{S}(\rho_1)$ *và* $\psi_2 = q_2(\rho_2) \vdash_n \mathcal{S}(\rho_2)$ *thì* $\psi = q(\rho) \vdash_n \mathcal{S}(\rho)$*, trong đó*
  (i) $q(\rho) = \exists \rho_1, \rho_2 \bullet (q_1(\rho_1) \wedge q_2(\rho_2) \wedge \rho|_u = \rho|_{u_1} \oplus \rho|_{u_2} \wedge \rho|_c = \rho|_{c_1} \oplus \rho|_{c_2}$*, và*
  (ii) $\mathcal{S}(\rho) = \forall \rho_1, \rho_2 \bullet (\rho|_u = \rho|_{u_1} \oplus \rho|_{u_2} \wedge \rho|_c = \rho|_{c_1} \oplus \rho|_{c_2} \wedge q(\rho) \wedge q(\rho_2) \Rightarrow \exists d_1, d_2 \bullet (d = max(d_1, d_2) \wedge \mathcal{S}(\rho_1) \wedge \mathcal{S}(\rho_2)))$

Trong định nghĩa này, biểu thức $q(\rho)$ *đúng khi và chỉ khi tồn tại một bộ phận* u, c *trong* ρ_1 *và* ρ_2 *sao cho* $q_1(\rho_1)$ *và* $q_2(\rho_2)$ *đúng. Tương tự như vậy* $\mathcal{S}(\rho)$ *đúng khi và chỉ khi bất kỳ một bộ phận* $\rho|_u$ *nào trong* $\rho_1|_u$*,* $\rho_2|_u$*, và* $\rho|_c$ *nào trong* $\rho|_c$*,* $\rho_2|_c$ *sao cho nếu cả* $q_1(\rho_1)$ *và* $q_2(\rho_2)$ *đúng và có một* bộ phận $d = max(d_1, d_2)$ *đúng.*

Định nghĩa 6.5 (Tương đương sử dụng tài nguyên)**.** *Hai thiết kế* $D = (\vartheta, \xi, \psi, d, \rho)$ *và* $D' = (\vartheta, \xi, \psi', d, \rho')$ *tương đương nhau về mặt tài nguyên khi và chỉ khi* $q \Rightarrow q'$ *và* $q' \Rightarrow q$ *và với mọi thành phần thứ* i *trong* ρ*,* ρ' *thì* $\rho = \rho'$.

6.4 Hợp đồng thời gian-tài nguyên

Định nghĩa 6.6 (Giao diện hợp đồng)**.** *Một giao diện của hợp đồng là một bộ* $\mathcal{I} \widehat{=} \langle \mathcal{F}d, Md_p, Md_r \rangle$ *trong đó:*

- $\mathcal{F}d$ *là tập hữu hạn các thuộc tính.*
- Md_p *và* Md_r *là sự khai báo tập các phương thức. Mỗi phương thức trong* Md_p *và* Md_r *có dạng* $op(X, Y)$*, tương ứng với từng thiết kế* $D = \langle \vartheta, \xi, \psi, d, \rho \rangle$ *đã cho, trong đó* X *là tập các biến đầu vào và* Y *là tập các biến đầu ra.* Md_p *là tập các phương thức cung cấp,* Md_r *là tập các phương thức yêu cầu,* $Md_p \cap Md_r = \emptyset$*.*

Từ giao diện của hợp đồng, ta có một hợp đồng được định nghĩa như sau:

Định nghĩa 6.7 (Hợp đồng thời gian-tài nguyên)**.** *Một hợp đồng là một bộ* $\mathfrak{C} = \langle \mathcal{I}, I, R_d, M_{Spec}, \mathcal{I}nv, \mathcal{I}nv_{R_d}, \wp \rangle$*, trong đó*

- $\mathcal{I} = (\mathcal{I}_p, \mathcal{I}_r)$ *là một giao diện. Đặt* $Md = Md_p \cup Md_r$*,* $\mathcal{F}d = \mathcal{F}d_p \cup \mathcal{F}d_r$*.*
- I *là sự khởi tạo các giá trị ban đầu cho từng thuộc tính trong tập* $\mathcal{F}d$*.*
- R_d *là sự khai báo tài nguyên, các giá trị này được nhập từ môi trường sử dụng hợp đồng.*
- M_{Spec} *là đặc tả phương thức, chúng liên quan đến từng phương thức* $op(X, Y)$ *trong tập* $Md = Md_p \cup Md_r$ *tương ứng với từng* thiết kế $D = \langle \vartheta, \xi, \psi, d, \rho \rangle$*.*
- $\mathcal{I}nv$ *là tập các* ràng buộc bất biến của hợp đồng *được biểu diễn bằng cặp* $(\mathcal{I}nv_p, \mathcal{I}nv_r)$*, trong đó* $\mathcal{I}nv_p$ *và* $\mathcal{I}nv_r$ *là công thức logic thời gian tuyến tính LTL ràng buộc tương ứng trên tập thuộc tính cung cấp và yêu cầu.*
- $\mathcal{I}nv_{R_d}$ *cho biết các điều kiện tài nguyên dưới dạng dịch vụ của thành phần có thể thực thi được, được thỏa bởi* R_d*.*
- $\wp$ *là một thể thức tương tác thời gian thực mô tả thể thức thực thi các phương thức có trong thành phần phần mềm, các thành phần phần mềm được phép thực thi đồng thời tại trạng thái đang xét của tài nguyên.*

Trong một số tình huống, các hợp đồng có thể được thay thế nhau nhằm tìm ra các tối ưu cho việc sử dụng các hợp đồng miễn là các hợp đồng được hiểu theo nghĩa cung cấp các dịch vụ tốt hơn nhưng tiêu thụ tài nguyên ít hơn.

Định nghĩa 6.8 (Làm mịn hợp đồng)**.** *Hợp đồng* $\mathfrak{C}_1 = \langle \mathcal{I}_1, I_1, R_{d_1}, M_{Spec_1}, \mathcal{I}nv_1, \mathcal{I}nv_{R_{d_1}}, \wp_1 \rangle$ *được gọi là làm mịn bởi hợp đồng* $\mathfrak{C}_2 = \langle \mathcal{I}_2, I_2, R_{d_2}, M_{Spec_2}, \mathcal{I}nv_2, \mathcal{I}nv_{R_{d_2}}, \wp_2 \rangle$*, ký hiệu* $\mathfrak{C}_1 \sqsubseteq \mathfrak{C}_2$ *khi và chỉ khi:*

- $\mathcal{F}d_1 \subseteq \mathcal{F}d_2, R_{d_1} \subseteq R_{d_2}$*, và* $I_2|_{\mathcal{F}d_1} = I_1|_{\mathcal{F}d_1}$*,* $I_2|_{R_{d_1}} \leq I_1|_{R_{d_1}}$ *(trong đó đối với các hàm* $f, f1, f2$ *và một tập* A*,* $f|_A$ *biểu thị sự hạn chế của* f *trên* A *biểu thức* $f_1 \leq f_2$ *cho biết* f_1, f_2 *cùng một miền giá trị* $f_1(\ddot{a}) \leq f_2(\ddot{a})$ *với mọi* $\ddot{a}$ *trong miền của chúng.*
- $Md_{p_1} \subseteq Md_{p_2}$*,* $Md_{r_2} \subseteq Md_{r_1}$
- *Với mọi phương thức op được khai báo trong* Md_{p_1} *thì* $M_{Spec_1}(op) \subseteq M_{Spec_2}(op)$

- *Với mọi phương thức op được khai báo trong* Md_{r_2} *thì* $M_{Spec_2}(op) \subseteq M_{Spec_1}(op)$
- $\mathcal{I}nv_2 \Rightarrow \mathcal{I}nv_1$.
- *Sự bất biến tài nguyên bị yếu dưới phép làm mịn, tức là* $\mathcal{I}nv_{R_{d_1}} \Rightarrow \mathcal{I}nv_{R_{d_2}}|_{R_{d_1}}$.
- $\wp_1|_{Md_{p_1}} \subseteq \wp_2|_{Md_{p_1}}$ *và* $\wp_2|_{Md_{r_2}} \subseteq \wp_1|_{Md_{r_2}}$.

Nói một cách đơn giản, trong hợp đồng $\mathfrak{C}_2$, nhiều dịch vụ và chức năng với chất lượng tốt hơn, và ít đòi hỏi tài nguyên và yêu cầu từ môi trường.

Định nghĩa 6.9 (Độ tương thích của các hợp đồng)**.** *Đặt* $\mathfrak{C}_1 = \langle \mathcal{I}_1, I_1, R_{d_1}, M_{Spec_1}, \mathcal{I}nv_1, \mathcal{I}nv_{R_{d_1}}, \wp_1 \rangle$ *và* $\mathfrak{C}_2 = \langle \mathcal{I}_2, I_2, R_{d_2}, M_{Spec_2}, \mathcal{I}nv_2, \mathcal{I}nv_{R_{d_2}}, \wp_2 \rangle$ *là hai hợp đồng.* $\mathfrak{C}_1$ *tương thích với* $\mathfrak{C}_2$ *khi và chỉ khi:*

- *Tập các thuộc tính và các phương thức cung cấp của chúng tương thích nhau. Tức là* $f \in \mathcal{F}d_1 \cap \mathcal{F}d_2$ *suy ra* $I_1(f) = I_2(f)$ *và* $op \in Md_{p_1} \cap Md_{p_2}$ *suy ra* $M_{Spec_1}(op) \Leftrightarrow M_{Spec_2}(op)$.
- *Tập các phương thức cung cấp và tập các phương thức yêu cầu tương thích nhau (đối với kết nối): Với mọi* $op \in Md_{r_1} \cap Md_{p_2}$*, điều này đúng với* $M_{Spec_1}(op) \sqsubseteq M_{Spec_2}(op)$ *và với mọi* $op \in M_{r_2} \cap Md_{p_1}$*, điều này đúng với* $M_{Spec_2}(op) \sqsubseteq M_{Spec_1}(op)$.

Định nghĩa 6.10 (Ghép song song thể thức thời gian-tài nguyên)**.** *Cho hai hợp đồng có khả năng ghép được* $\mathfrak{C}_1 = \langle \mathcal{I}_1, I_1, R_{d_1}, M_{Spec_1}, \mathcal{I}nv_1, \mathcal{I}nv_{R_{d_1}}, \wp_1 \rangle$ *và* $\mathfrak{C}_2 = \langle \mathcal{I}_2, I_2, R_{d_2}, M_{Spec_2}, \mathcal{I}nv_2, \mathcal{I}nv_{R_{d_2}}, \wp_2 \rangle$*. Phép ghép song song của hai hợp đồng* $\mathfrak{C}_1$ *và* $\mathfrak{C}_2$ *ký hiệu bởi* $\mathfrak{C}_{||} = \mathfrak{C}_1||\mathfrak{C}_2$ *là một hợp đồng* $\mathfrak{C} = \langle \mathcal{I}, I, R_d, M_{Spec}, \mathcal{I}nv, \mathcal{I}nv_{R_d}, \wp \rangle$ *các thành phần trong hợp đồng mới được tính như sau:*

- $\mathcal{I} = (\mathcal{I}_{p1} \cup \mathcal{I}_{p2}, \mathcal{I}_{r1} \cup \mathcal{I}_{r2})$
- $I = I_1 \cup I_2$
- $R_d = R_{d_1} \oplus R_{d_2}$
- $M_{Spec} = M_{Spec_1} \cup M_{Spec_2}$*, trong đó* $Md_r = Md_{r1} \cup Md_{r2}$, $Md_p = Md_{p1} \cup Md_{p2}$
- $\mathcal{I}nv$ *được tính như sau:* $\mathcal{I}nv = \mathcal{I}nv_1 \wedge \mathcal{I}nv_2$
- $\mathcal{I}nv_{R_d} = \mathcal{I}nv_{R_{d_1}} \uplus \mathcal{I}nv_{R_{d_2}}$
- $\wp = \wp_1||\wp_2$

Luận án sử dụng toán tử $\uplus$ để kết hợp các thành phần trong các thành phần bất biến tài nguyên, và toán tử có ngữ nghĩa như sau: Giả sử $\varrho, \varrho_1, \varrho_2$ là các điều kiện tài nguyên, $\mathcal{I}nv_{R_d}(\varrho) = \mathcal{I}nv_{R_{d_1}}(\varrho_1) \uplus \mathcal{I}nv_{R_{d_2}}(\varrho_2) \mathrel{\widehat{=}} \exists \varrho_1, \varrho_2 \bullet (\varrho|_c = \varrho_1|_c + \varrho_2|_c \wedge \varrho|_u = \varrho_1|_u + \varrho_2|_u)$.

Ngoài cách ghép song song, chúng ta có thể ghép các hợp đồng bằng phương pháp ghép nối tiếp.

Định nghĩa 6.11 (Ghép nối tiếp hợp đồng thời gian-tài nguyên)**.** *Cho hai hợp*

đồng có khả năng ghép được $\mathfrak{C}_1 = \langle \mathcal{I}_1, I_1, R_{d_1}, M_{Spec_1}, \mathcal{I}nv_1, \mathcal{I}nv_{R_{d_1}}, \wp_1 \rangle$ *và hợp đồng* $\mathfrak{C}_2 = \langle \mathcal{I}_2, I_2, R_{d_2}, M_{Spec_2}, \mathcal{I}nv_2, \mathcal{I}nv_{R_{d_2}}, \wp_2 \rangle$. *Phép ghép nối tiếp của* $\mathfrak{C}_1$ *với* $\mathfrak{C}_2$, *ký hiệu bởi* $\mathfrak{C}_\circ = \mathfrak{C}_1 \circ \mathfrak{C}_2$, *là một hợp đồng* $\mathfrak{C}_\circ = \langle \mathcal{I}, I, R_d, M_{Spec}, \mathcal{I}nv, \mathcal{I}nv_{R_d}, \wp \rangle$, *các thành phần trong hợp đồng mới được tính như sau:*

- $\mathcal{I} = (\mathcal{I}_{p1} \cup \mathcal{I}_{p2}, \mathcal{I}_{r1} \setminus \mathcal{I}_{r1}|_{Md_{r_1} \cap Md_{p_2}} \cup \mathcal{I}_{r2})$
- $I = I_1 \setminus I_1|_{\mathcal{F}d_{r_1} \cap \mathcal{F}d_{p_2}} \cup I_2$
- $R_d = R_{d_1} \oplus R_{d_2}$
- $M_{Spec} = M_{Spec_1} \cup M_{Spec_2}$, *các thành phần trong* Md *được tính như sau:* $Md_r = Md_{r1} \cup Md_{r2}$, $Md_p = Md_{p1} \cup Md_{p2}$.
- $\mathcal{I}nv$ *được tính như sau:* $\mathcal{I}nv_p = \mathcal{I}nv_{p_1} \wedge \mathcal{I}nv_{p_2}$, $\mathcal{I}nv_r = \mathcal{I}nv_{r_1} \wedge \mathcal{I}nv_{r_2}$.
- $\mathcal{I}nv_{R_d} = \mathcal{I}nv_{R_{d_1}} \uplus \mathcal{I}nv_{R_{d_2}}$
- $\wp = \wp_1 \circ \wp_2$

Định lý 6.1. *Cho ba hợp đồng* $\mathfrak{C}_1, \mathfrak{C}_2$ *và* $\mathfrak{C}_3$. *Ta có*

(i) $\mathfrak{C}_1 || \mathfrak{C}_2) = \mathfrak{C}_2 || \mathfrak{C}_1)$
(ii) $(\mathfrak{C}_1 || \mathfrak{C}_2) || \mathfrak{C}_3 = \mathfrak{C}_1 || (\mathfrak{C}_2 || \mathfrak{C}_3)$
(iii) $(\mathfrak{C}_1 ._\theta \mathfrak{C}_2) ._\theta \mathfrak{C}_3 = \mathfrak{C}_1 ._\theta (\mathfrak{C}_2 ._\theta \mathfrak{C}_3)$

Định nghĩa 6.12 (Phép cắm đầy đủ)**.** *Cho hai hợp đồng* $\mathfrak{C}_1 = \langle \mathcal{I}_1, I_1, R_{d_1}, M_{Spec_1}, \mathcal{I}nv_1, \mathcal{I}nv_{R_{d_1}}, \wp_1 \rangle$ *và* $\mathfrak{C}_2 = \langle \mathcal{I}_2, I_2, R_{d_2}, M_{Spec_2}, \mathcal{I}nv_2, \mathcal{I}nv_{R_{d_2}}, \wp_2 \rangle$, *một phép cắm đầy đủ của hợp đồng* $\mathfrak{C}_1$ *và* $\mathfrak{C}_2$ *là hợp đồng* $\mathfrak{C} = \langle \mathcal{I}, I, R_d, M_{Spec}, \mathcal{I}nv, \mathcal{I}nv_{R_d}, \wp \rangle$, *ký hiệu bằng* $\mathfrak{C} \cong \mathfrak{C}_1 \gg \mathfrak{C}_2$, *được định nghĩa như sau:*

- $\mathcal{I} = (\mathcal{I}_{p_1} \cup \mathcal{I}_{p_2}, \mathcal{I}_{r_2})$.
- $R = R_2 \oplus R_1$.
- $M_{Spec} = M_{Spec_1} \cup M_{Spec_2}$.
- $I = I_1|_{\mathcal{F}d_{r_1}} \cup I_2$.
- $\mathcal{I}nv = \mathcal{I}nv_1 \wedge \mathcal{I}nv_2$.
- $\mathcal{I}nv_{R_d} = \mathcal{I}nv_{R_{d_1}} \uplus \mathcal{I}nv_{R_{d_2}}$.
- $\wp = \wp_1 \cup \wp_2 \cup \wp_1 || (\wp_2 \cap \{op \in Md_{p_2} \setminus Md_{r_1}\}^*)$.

6.5 Hệ thống thời gian-tài nguyên

Định nghĩa 6.13 (Thành phần phần mềm thụ động)**.** *Thành phần phần mềm thụ động thời gian thực là một bộ* $P_{\mathbb{C}} \cong \langle \mathfrak{C}, Mcode \rangle$, *trong đó* $\mathbb{C}$ *đồng nhất với tên của thành phần, bao gồm*

- *Một hợp đồng* $\mathfrak{C} = \langle \mathcal{I}, I, R_d, M_{Spec}, \mathcal{I}nv, \mathcal{I}nv_{R_d}, \wp \rangle$ *được thực thi bởi* $P_{\mathbb{C}}$.
- *Một ánh xạ* $Mcode$ *gán mỗi phương thức* op *trong* Md_p *một thiết kế được xây dựng từ các toán tử cơ bản, với giả định sử dụng và tiêu thụ một lượng tài nguyên, và các lời gọi tới các phương thức* m *trong* Md_r. $Mcode$ *thỏa* $M_{Spec}(op) \sqsubseteq Mcode(op)$ *đối với tất cả phương thức* op *trong* Md_p.

Định lý 6.2. *Đặt* $P_{\mathbb{C}_i} = \langle \mathfrak{C}_i, Mcode_i \rangle$, *i=1,2 là thành phần phần mềm thụ động.* $P_{\mathbb{C}_1}$ *được gọi là làm mịn bởi* $P_{\mathbb{C}_2}$, *ký hiệu* $P_{\mathbb{C}_1} \sqsubseteq P_{\mathbb{C}_2}$ *khi và chỉ khi* $\mathfrak{C}_1 \sqsubseteq \mathfrak{C}_2$.

Định nghĩa 6.14 (Giao diện của hệ thống)**.** *Giao diện hệ thống là một bộ* $\mathcal{I}_S = \langle Evt, Fd, SMd_p \rangle$*, trong đó* SMd_p *là tập hữu hạn các phương thức,* Fd *là một tập hữu hạn các thuộc tính,* Evt *là tập hữu hạn các sự kiện.*

Định nghĩa 6.15 (Hợp đồng của hệ thống)**.** *Hợp đồng hệ thống là một bộ* $Sys\mathfrak{C} = \langle \mathcal{I}_s, SMSpec, Inv, Behav \rangle$*, trong đó*

- $I_s = \langle Evt, Fd, SMd_p \rangle$ *là giao diện hệ thống.*
- $SMSpec$ *là đặc tả phương thức liên quan đến mỗi phương thức* $op(X, Y)$ *trong* SMd_p *với một thiết kế* $D(op) = (\vartheta, \xi, \psi, d, \rho)$*, và*
- $Behav$ *là sự mô tả hành vi bên ngoài là một tập hữu hạn của* $\{evt, m | evt \in Evt, m \in SMd_p\}^*$*. Mỗi hành vi của* $Behav$ *được gọi đặc tả tiến trình.*

Định nghĩa 6.16 (Thành phần phần mềm chủ động)**.** *Thành phần phần mềm chủ động là một bộ* $A_{\mathbb{C}} = \langle \mathfrak{C}, Sys\mathfrak{C}, Mcode \rangle$ *bao gồm*

- *Một hợp đồng* $\mathfrak{C}$ *với tập giao diện cung cấp rỗng* $\mathcal{I}_p = \langle \emptyset, \emptyset \rangle$*.*
- *Một hợp đồng hệ thống* $Sys\mathfrak{C} = \langle I_s, SMSpec, Inv, Behav \rangle$*.*
- *Các tiến trình thực thi gọi các dịch vụ trong* $Mcode$ *gán mỗi phương thức* op *trong* SMd_p *một thiết kế được xây dựng từ các toán tử cơ sở. Điều kiện sau sẽ thỏa bởi* $Mcode : (SMSpec(op) \sqsubseteq Mcode(op))$ *với mọi* $op \in SMd_p$*.*

Một hệ thống trong mô hình thành phần của luận án là một *Thành phần phần mềm chủ động* cắm vào một *Thành phần phần mềm thụ động đóng.*

Định nghĩa 6.17 (Hệ thống)**.** *Hệ thống là một cặp thành phần phần mềm chủ động* $A_{\mathbb{C}} = \langle \mathfrak{C}, Sys\mathfrak{C}, Mcode \rangle$ *và một thành phần phần mềm thụ động* $P_{\mathbb{C}} = \langle \mathfrak{C}', Mcode' \rangle$*, sao cho* $\mathfrak{C} \gg \mathfrak{C}'$*.*

6.6 Ngôn ngữ đặc tả thời gian thực mẫu

6.6.1 Mô tả phương thức

Một thiết kế thời gian có ràng buộc các thành phần phi chức năng được đặc tả bằng từ khóa **method**, các phương thức ở đây hoặc là loại **provided** hoặc là **required**, tên phương thức được đặt bằng từ khóa **name**

```
1 <[provided][required] > method {
2 name <Name of method>;
3 resource r = Type of resource;
4 duration int d = constant;
5 assert <Predicate>;
6 specification Predicate;
7 }
```

6.6.2 Mô tả hợp đồng

```
1  component <Component name>{
2    provided feature <Type List of features>;
3    required feature  <Type List of features>;
```

```
4     system resource <Resource feature>;
5     invariant <LTL formulae>;
6     resource invariant <Predicates>;
7      <[provided][required] > method {
8        name <Name of method>;
9        resource r = Type of resource;
10       duration int d = constant;
11       specification Predicate;
12     }
13    protocol <List of Regular Expression>;
14    code {Code};
15   }
```

6.6.3 Mô tả thành phần phần mềm chủ động

Mặc định hợp đồng là thuộc loại *Passive component*, nếu thành phần phần mềm thuộc loại *Active component* phải thêm từ khóa **active** vào trước từ khóa **component**. Trong *Active component*

```
1  active component <Component name>{
2    required feature  <Type List of features>;
3    system resource <Resource feature>;
4    invariant <LTL formulae>;
5    system assert  <Predicate>;
6    resource invariant <Predicates>;
7     <required> method {
8        name <Name of method>;
9        resource r = Type of resource;
10       duration int d = constant;
11       assert  <Predicate>;
12       specification Predicate;
13     }
14    protocol <List of Regular Expressions>;
15    code {Code};
16   }
```

6.6.4 Mô tả hệ thống

Hệ thống được tạo bởi thành phần phần mềm chủ động và thành phần phần mềm thụ động.

6.7 Tổng kết chương

Chương này luận án mở rộng thiết kế thời gian bằng cách bổ sung các đặc tả phi chức năng trên hai khía cạnh gồm bổ sung đặc tả phi chức năng cho thiết kế thời gian trở thành thiết kế tài nguyên-thời gian và bổ sung ràng buộc phi chức năng vào hợp đồng nhằm lập luận cho việc cung cấp tài nguyên cũng như sử dụng tài nguyên của hệ thống thời gian thực dựa trên thành phần.

Chương 7

Kết luận

7.1 Các kết quả đạt được

Qua quá trình nghiên cứu luận án đã đạt được một số kết quả chính sau đây: Thứ nhất, luận án đề xuất mô hình cho thành phần phần mềm thời gian thực với thể thức tương tác tương tranh thời gian và thể thức tương tranh thời gian có ràng buộc tài nguyên, và sử dụng ôtômát thời gian và ôtômát trọng số để mô hình hóa các chuỗi hành vi của môi trường. Trên cơ sở đó, luận án đề xuất các thuật toán kiểm tra sự tuân thủ các chuỗi hành vi của môi trường với thể thức tương tác của thành phần phần mềm thời gian thực trên cả hai khía cạnh chức năng và phi chức năng. Thứ hai, luận án mở rộng lý thuyết giao diện thành phần trở thành lý thuyết giao diện thành phần thời gian thực để đặc tả, mô hình hóa giao diện và kiểm tra các tính chất của hệ thống thời gian thực dựa trên thành phần. Thứ ba, luận án đề xuất mở rộng lý thuyết hợp đồng thành hợp đồng thời gian thực để đặc tả, kiểm chứng thành phần phần mềm trong hệ thống thời gian thực dựa trên thành phần trên cả hai khía cạnh chức năng và phi chức năng. Trong đóng góp này, luận án cũng bổ sung ngôn ngữ thời gian thực mẫu nhằm tăng khả năng ứng dụng các kết quả nghiên cứu lý thuyết vào thực tế.

7.2 Hướng phát triển tiếp theo

Dưới đây luận án tóm tắt một số vấn đề có thể nghiên cứu trong tương lai: Trong đóng góp thứ nhất, luận án đang nghiên cứu mở rộng mô hình thời gian thực cho hệ thống có quy mô lớn trên thực tế như hệ thống nhúng, hệ thống tác tử di động và triển khai cho các dịch vụ web trên cơ sở các thể thức tương tác tương tranh thời gian có ràng buộc tài nguyên. Trong đóng góp thứ hai, khoảng trống giữa lý thuyết giao diện thành phần, Assume/Guarantee và lý thuyết hợp đồng cần được nghiên cứu để phân tích đánh giá ưu nhược điểm khi sử dụng các lý thuyết này áp dụng vào phát triển phần mềm. Trong đóng góp thứ ba, chưa có một công cụ nào để có thể áp dụng lý thuyết này nên công việc tiếp theo cần phát triển bộ công cụ phát triển phần mềm cho hệ thống thời gian thực dựa trên lý thuyết đã đề xuất.

DANH MỤC CÁC CÔNG TRÌNH KHOA HỌC CỦA TÁC GIẢ LIÊN QUAN ĐẾN LUẬN ÁN

[1] Nguyen Trinh Dong, Dang Van Hung, Truong Anh Hoang (2011),"Real-Time Relational Interface Behavior Modeling and Specification", *KSE 2011-Third International Conference on Knowledge and Systems Engineering, pp.112-119.*

[2] Nguyen Trinh Dong (2015), "Memory Resource Estimation of Component-Based Systems", *4th International Conference, ICCASA 2015, Vung Tau, Vietnam, November 26-27, pages~73-82.*

[3] Dang Van Hung, Nguyen Trinh Dong and Truong Anh Hoang (2017), "A Model for Real-timed Concurrent Interaction Protocols in Component Interfaces", *VNU Journal of Science: Computer Science and Communication Engineering, vol.33, no.1, pp.*

[4] Nguyen Trinh Dong, Dang Van Hung, Truong Anh Hoang (2017), "A formal contract-based model for component-based real-time systems", NICS 2017 - 3rd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS), p--, (Accepted)

[5] Nguyen Trinh Dong (2017), "A General Model for Quality Analyzing of Functional and Non-functional Features in Real-Time Systems", *The 6th Conference on Information Technology and Its application, (CITA 2017)*, Da Nang, pages~--, (Submitted)