

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成 24 年 3 月 31 日現在

機関番号：33917
 研究種目：基盤研究（C）
 研究期間：2009～2011
 課題番号：21500042
 研究課題名（和文） プロダクトラインソフトウェア工学における要求とアーキテクチャとの関連付け
 研究課題名（英文） On tractability from requirements to architecture in product line software engineering
 研究代表者
 野呂 昌満（NORO MASAMI）
 南山大学・情報理工学部・教授
 研究者番号：40189452

研究成果の概要（和文）： 同一の応用領域に属する製品に共通する特性を構造的に把握し、それらの特性に合った技術を体系的に応用する PLSE(プロダクトラインソフトウェア工学)が生産性向上に有効であるとして提案されている。本研究では、非機能特性と機能特性の両方を適切にモデル化したプロダクトラインアーキテクチャ（製品系列で共通に用いられるアーキテクチャ）を系統立てて構築する方法ならびに要求との追跡性を保証する方法の提案し、検証した。それらの方法を基礎として、アーキテクチャレベルでの誤り検出方法の応用、ソースプログラムの欠陥検出や書換えを行うためのツール基盤の整備を行い、PLSE の実用化について考察した。

研究成果の概要（英文）： PLSE(Product Line Software Engineering) has been expected to be another breakthrough to the high productivity. We proposed and validated the methods for constructing product line architecture allowing for non-functional as well as functional requirements, and for ensuring tractability between requirements for the software and the architecture. Based on the methods, we also discussed about the issues on architecture-level pre-execution check, code inspection, and tools for generating codes from the architecture.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2009年度	1,500,000	450,000	1,950,000
2010年度	900,000	270,000	1,170,000
2011年度	1,000,000	300,000	1,300,000
年度			
年度			
総計	3,400,000	1,020,000	4,420,000

研究分野：総合領域

科研費の分科・細目：情報学・ソフトウェア

キーワード：PLSE モデル駆動開発 アスペクト指向 実証的手法 形式的仕様記述

1. 研究開始当初の背景

ソフトウェア開発における生産性の向上には、同一の応用領域に属する製品に共通する特性を構造的に把握し、それらの特性に合った技術を体系的に利用することが有効である。この考えに基づき、プロダクトラインソ

フトウェア工学（Product Line Software Engineering 以下 PLSE）が提案されている。PLSE の実践に向けては、コア資産相互、さらにコア資産と個別プロダクトの間の追跡性をいかに管理するかが重要である。研究代表者および分担者はこれまでにソフトウェア

アーキテクチャと再利用部品との追跡性管理や個別プロダクトの自動生成などについて研究を行い、おもに下流工程部分の実用化の基礎を与えた。一方で、上流工程、すなわち仕様モデルとソフトウェアアーキテクチャとの間の追跡性に関しては十分に体系化が行われていない。アーキテクチャ設計の成否は設計者の経験に依存しているのが現状である。これは、プロダクトラインに対する要求には本質的に多様な視点が含まれ、それらを素直にアーキテクチャに対応付けるための記述体系や分析方法が確立できていないことに起因する。研究代表者と分担者らは、組込みシステムを対象としたアーキテクチャ設計やアーキテクチャスタイルに関する研究を通じ、アスペクト指向の概念に基づく要求とアーキテクチャとの構造的な対応付けの手法を提案してきた。この手法に基づくコア資産間の追跡方法や支援技術を整理・実現することにより、この問題を解決できるという着想に至った。

2. 研究の目的

本研究は、プロダクトラインソフトウェア開発における、要求とソフトウェアアーキテクチャとの意味的な関連の定式化を目的とする。そのために、本研究では、アスペクト指向に基づくソフトウェアアーキテクチャのモデル化を技術的な中核と位置づけ、多視点からの要求分析とモデル化の方式、さらにアスペクト指向に基づくモデル変換論理をプロダクトライン開発支援のための基盤として確立する。これらの基盤技術を具体化した開発支援ツールとともに、実証的アプローチに基づき開発改善のプロセスを具体化することで、プロダクトライン要求に合わせて継続的に発展可能な統合支援環境を実現する。支援対象の一例としてロボットなどの組込みソフトウェアを取り上げ、プロダクトラインソフトウェア開発の実践に提案する手法やその実現である支援環境が有効であることを考察する。

3. 研究の方法

つぎに示す4項目を研究副課題として研究を行った。

(1) 要求からアーキテクチャへのモデル変換方式として、非機能要求とアーキテクチャスタイルの関連を整理し、アーキテクチャの設計と文書化を系統立てて行う方法について研究した。我々はこれまでに組込みシステムのためのアスペクト指向ソフトウェアアーキテクチャスタイル E-AoSAS++ を提案してきた。デザインパターンを手掛かりとして、追跡性を確保した E-AoSAS++ によるアーキテクチャの設計方法の検討を行った。

(2) 要求とアーキテクチャのモデル記述方式として、E-AoSAS++ の記法と操作的意味について形式化を行い、アーキテクチャレベルの形式仕様記述に対するモデル検査の実用化について研究を行った。

(3) ツールによる開発支援方式として、我々のこれまでの成果であるソフトウェアリポジトリや、XML処理系およびその効率化に関する研究成果に基づき、ソースプログラムの欠陥の検出や自動書換えを行うためのツール基盤の整備、大量のソフトウェア資産を効率的に検索、処理するためのXQuery処理系の開発を行った。

(4) 実証的アプローチによるプロダクトライン開発の支援として、アスペクト指向により横断的関心事としてモジュール化されたコンポーネントについて、実証的見地から評価を行った。

4. 研究成果

研究の方法で挙げた4つの研究副課題と関連づけて成果を述べる。

(1) 要求からアーキテクチャへのモデル変換方式として、非機能要求とアーキテクチャスタイルの関連を定義したアーキテクチャデザインマップ（以下、デザインマップ）の提案を行った（雑誌論文[1]、学会発表[8]）。プロダクトライン開発では、要求からアーキテクチャ内のコンポーネントが追跡可能である事が前提である。とくに、非機能要求とプロダクトラインアーキテクチャ（製品系列で共通に用いられるアーキテクチャ）の関係についてはよく理解されていなかった。デザインマップを用いることで、プロダクトラインアーキテクチャの設計と文書化を系統立てて行うことが可能となる。

デザインマップのメタモデルを図1に示す。デザインマップは ISO9126 で定義された非機能要求と Clements らによって提案されたコンポーネント・コネクタ・アーキテクチャスタイル(以下、C&Cスタイル)との関連を定義する。

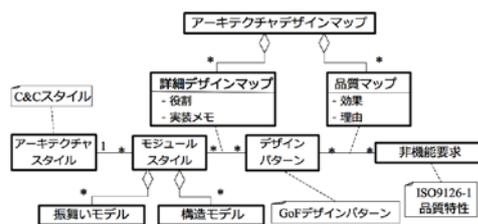


図1 デザインマップのメタモデル

非機能要求とアーキテクチャスタイルの

対応づけにおいてギャップが存在するが、本研究ではデザインパターンを用いてそのギャップを埋めた。デザインパターンはアーキテクチャスタイルの実装と非機能要求の実現の2つの役割をもち、デザインマップではそれぞれ、詳細デザインマップと品質マップとして表現される

詳細デザインマップは、アーキテクチャスタイル、モジュールスタイル、デザインパターンの対応関係を定義する。モジュールスタイルは、パイプ&フィルタやクライアント・サーバなどのC&Cスタイルに対してデザインパターンを適用して、より詳細化したものである。モジュールスタイルは構造モデルと振舞いモデルから構成される(図2)。

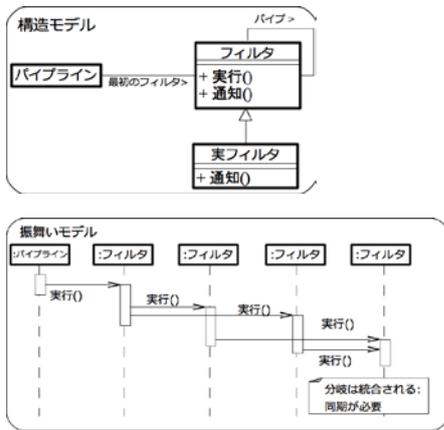


図2 構造モデルと振舞いモデル

品質マップではデザインパターンと非機能要求の対応関係を定義した。パターンを適用することで、改善される非機能要求(例えば、実行時間が減少する)、悪化する非機能要求(例えば、実行時間が増加する)、どちらにもなりうる非機能要求などを理由とともに整理した。

デザインマップを用いたアーキテクチャの設計プロセスを図3に示す。4つのプロセスから構成される。

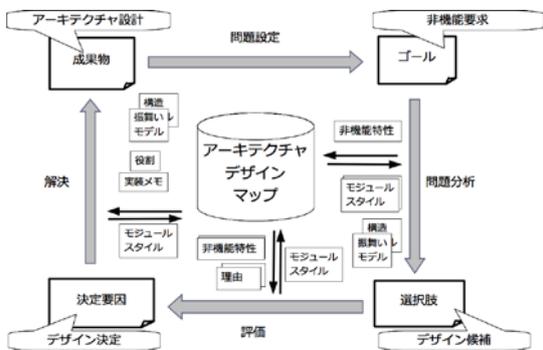


図3 アーキテクチャデザインプロセス

1. 問題設定: 満たすべき非機能要求などのゴールを設定する。
2. 問題分析: デザインマップを利用し、非機能要求を満たすアーキテクチャ設計上の選択肢を抽出する。
3. 評価: 問題分析で得られた選択肢から設計の決定をする。デザインマップで定義された非機能要求の理由などが決定作業を支援し、決定要因なども明確に文書化することができる(図4)。
4. 解決: 詳細デザインマップの構造モデルや振舞いモデルなどを参照し、アーキテクチャを設計する。

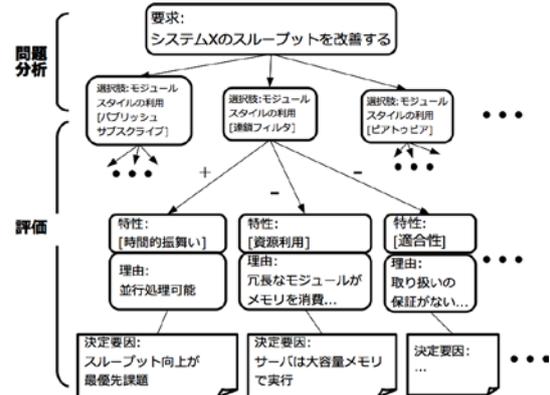


図4 アーキテクチャの文書化の例

情報処理学会組込みシステム研究会が主催するロボットチャレンジを題材に評価を行った。飛行船制御システムに対して提案したデザインマップを適用し、アーキテクチャの設計やその文書化に有効であったことを確認した。図5にデザインマップにより設計された飛行船の基地局制御のアーキテクチャを示す。

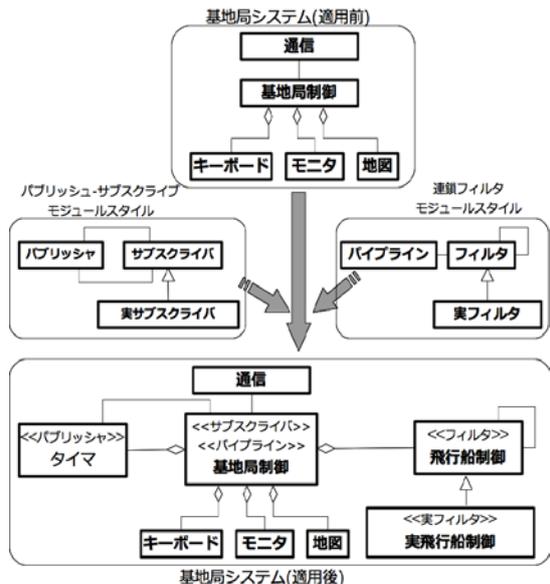


図5 飛行船の基地局制御のアーキテクチャ

今後の展望として、自動販売機などの様々な組込み機器にデザインマップを適用して評価すること、デザインマップからドキュメントの自動生成や管理などを行う支援ツールの開発などが挙げられる。

(2) 要求とアーキテクチャのモデル記述方式の成果として、つぎの2つが挙げられる(雑誌論文 [8][11]、学会発表[2][3][7])。

- ①. E-AoSAS++の記法と操作的意味の形式化
- ②. アーキテクチャレベルの形式仕様記述に対するモデル検査の適用方法の提案

①は、組込みシステムを並行状態遷移機械の集合として表現する E-AoSAS++によるアーキテクチャ記述を、静的モデルと動的モデルの観点から形式化した。静的モデルではアーキテクチャ記述の基本構成要素をコンポーネント、コンポーネントオブジェクト、アクション、状態遷移機械、ビューの5種類とし、それぞれのUMLによる表記とZ言語による表記を定めた。動的モデルでは並行実行モデルの操作的意味をZ言語により定義した。コンポーネントの動作をミリー型状態遷移機械で定義し、コンポーネント間のキューを介した非同期なイベントのやりとりによって計算が進行する。

②は、アーキテクチャレベルの形式仕様記述に対して、モデル検査を用いて振舞いを検証する方法を、つぎの2つの観点から研究した。

- (a) 状態爆発を回避するための検証の枠組み
- (b) モデル検査におけるフォールト検出作業の支援

モデル検査による振舞い検証を実用化する際の障壁として、状態爆発の問題がある。(a)では、複雑な制御を横断的な関心事としてモジュール化することで問題を局所化し、分割されたモジュールを合成する際に不整合が生じないことを検証する枠組みを提案した。簡単な事例に対して、モジュールの振舞いをプロセス代数CSPで記述し、CSPの代表的なモデル検査器FDRを用いて、提案した検証方法の有効性を確認した。

モデル検査では、デッドロックを検出した場合や仕様が満たされないことを検出した場合、反例が提示されるが、その原因となるフォールトを特定する作業は困難である。

(b)では、プロセス代数CSPを対象として、記述の誤りをパターンとして提示することにより、フォールト検出作業を軽減する方法を提案した。反例からフォールトを特定する作業はプログラミングにおけるデバッグ作業に相当する。プログラミングにおけるコードインスペクションツールのように、仕様記

述に対してフォールトの可能性のある部分を、フォールトパターンとして指摘することで検証コストを削減できる。共有資源におけるデッドロックなどの一般的な並行性に関するフォールトと、デザインパターンにおける振舞いなどのアプリケーション固有のフォールトについて分析し、パターンとして定義し、その有効性を確認した。

今後の展望として、モデル検査の実用化に向けて、提案した方法に基づいてCSP/FDRによるモデル検査を支援するツールの開発などが挙げられる。

(3) ツールによる開発支援方式の成果はつぎの2つである。

- ①. ソースプログラムの欠陥の検出や自動書換えを行うためのツール基盤の実現(雑誌論文[4][5][7][9]、学会発表[5][9])
- ②. XMLにより記述された大量のソフトウェア資産を効率的に検索、処理するためのXQuery処理系の開発(雑誌論文[2]、学会発表[8])

①では、OJL(On the Job Learning)と呼ぶ産学連携の開発プロジェクトにより、Javaソースコードを解析して欠陥を検出するインスペクションツールを設計・実現し、そのソフトウェアアーキテクチャ設計と進化について議論した。ソースコードの欠陥検査や品質向上に対する要求は多様であり、さまざまな要求に応じた変更やカスタマイズを可能とする柔軟性がインスペクションツールには必要となる。インスペクションツールの解析性、変更性、試験性、効率性を向上させるためのソフトウェアアーキテクチャを、GoFのデザインパターンを組み合わせることで設計した。ツールを実用していく上で、解析処理の高度化や、検査精度の向上、空間効率の改善など、さまざまな機能追加や保守を重ねてきた。本研究では、これら変更要求や機能追加要求への対応を通じ、デザインパターンを用いて設計したインスペクションツールのソフトウェアアーキテクチャが、変更の影響を最小にとどめながらソフトウェア構造の改善と洗練を行うための保守の基盤として有効に機能していることを示した。この研究で得られた知見は、(1)の要求からアーキテクチャへのモデル変換方式にも活用されている。

②では、XML文書の検索を省メモリかつ高速で実行するために、問い合わせをコンパイル方式で行うXQuery処理系の生成系を設計・実現した。コンパイル方式では、プログラム生成とコンパイルの手間が増えるが、あらかじめ質問式を解析することで、問い合わせに使用されるXML文書の情報だけを部分木

として作成することができ、高速で軽量の処理が可能となる。提案した生成系は、XQuery 質問式と DTD を入力とし、XML 文書の字句・構文解析と問い合わせ処理を行う lex, yacc プログラムを出力する。XML 文書中の異なる箇所からのデータの突き合わせを行う結合処理は XML 文書木に対して処理を行う必要があり、コンパイル方式の処理だけでは十分な高速化を行えない。本研究では結合処理を同値比較と大小比較に分類し、同値比較にはハッシュ法、大小比較には二分探索法を用いて高速化を行った。実現した生成系を用いて実験を行い、その有効性を評価した。必要な要素のみによる木を作成することでメモリ使用量を削減でき、結合処理を高速で処理可能であることを確認した。

今後の展望として、開発支援ツールの各機能をサービス化して連携させることなどが挙げられる。

(4) 実証的アプローチによるプロダクトライン開発の支援では、アスペクト指向技術を用いたリファクタリングの特性を分析した(雑誌論文[6][10])。2つの大規模ソフトウェアシステムに対して実施されたアスペクト指向技術を用いたリファクタリングプロジェクトを題材に、リファクタリングの前後でのコンポーネントの利用関係の変化やクローンの関係の変化を評価基準として分析した。結果として、アスペクト抽出により、全体的な複雑度は大きく変わらずに、関心事のみの分離ができており、一部の特定の種類のコンポーネントについては、被利用関係が非常に細かくなりやすいこと、クローンの関係はアスペクト中に拡散しやすいことなどが明らかになった。

今後の展望として、(1)で挙げた開発プロセスに対して実証的アプローチを適用し、作成されたプロダクトを計測する技術の確立、プロセスの改善などが挙げられる。

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文] (計 12 件)

[1] A. Sawada, M. Noro, H.-M Chang, Y. Hachisu, A. Yoshida, A Design Map for Recording Precise Architecture Decisions, Proc. the 18th Asia-Pacific Software Engineering Conference (APSEC2011), 査読有, 2011, pp. 298-305

[2] 蜂巢吉成, 野呂昌満, 沢田篤史, 張漢明, 吉田敦, コンパイル方式による XQuery 問い合わせプログラム生成方法, ソフトウェア工学の基礎 XVIII ー日本ソフトウェア科学会 FOSE 2011, 査読有, 2011, pp. 21-30

[3] 蜂巢吉成, 吉田敦, 野呂昌満, 沢田篤史, 張漢明, HTML 要素の状態を考慮した CSS の拡張方法の提案, 電子情報通信学会論文誌 D, 査読有, Vol. J94-D, No. 11, 2011, pp. 1931-1934

[4] 沢田篤史, 野呂昌満, 蜂巢吉成, 張漢明, 吉田敦, 長大介, 浦野彰彦, ソースコードインスペクションツールのためのソフトウェアアーキテクチャの設計と進化, 日本ソフトウェア科学会 編 コンピュータソフトウェア, 査読有, 28 巻 4 号, 2011, pp. 4_241-4_261

[5] 曾我展世, 吉田敦, 蜂巢吉成, 沢田篤史, 張漢明, 野呂昌満, 表現の違いを考慮したマクロ逆置換方法の提案, ソフトウェアエンジニアリングシンポジウム 2011 (SES2011) 論文集ー情報処理学会シンポジウムシリーズ, 査読有, 2011, 6 ページ

[6] R. Yokomori, H. Siy, N. Yoshida, M. Noro, K. Inoue, Measuring the Effects of Aspect-Oriented Refactoring on Component Relationships: Two Case Studies, Proceedings of the 10th Annual Aspect-Oriented Software Development Conference (AOSD2011), 査読有, 2011, pp. 215-226

[7] 浦野彰彦, 沢田篤史, 野呂昌満, 蜂巢吉成, 張漢明, 吉田敦, デザインパターンを用いたソースコードインスペクションツールのソフトウェアアーキテクチャ設計, ソフトウェア工学の基礎 XVIIー日本ソフトウェア科学会 FOSE2010, 査読有, 2010, pp. 15-24

[8] 張漢明, 野呂昌満, 沢田篤史, 蜂巢吉成, 吉田敦, モデル検査を用いた振舞検証の実用化技術に関する考察ー網羅性に注目してー, ソフトウェア工学の基礎 XVIIー日本ソフトウェア科学会 FOSE2010, 査読有, 2010, pp. 107-112

[9] 吉田敦, 蜂巢吉成, 沢田篤史, 張漢明, 野呂昌満, 属性付き字句系列に基づくプログラム書換え支援環境の試作, ソフトウェアエンジニアリング最前線ー情報処理学会 SES2010, 査読有, 2010, pp. 119-126

[10] R. Yokomori, H. Siy, M. Noro, K. Inoue, Assessing the Impact of Framework Changes Using Component Ranking, Proc. IEEE 25th International Conference on Software Maintenance (ICSM2009), 査読有, 2009, pp. 189-198

[11] 張漢明, 野呂昌満, 沢田篤史, 蜂巢吉成, アスペクト指向ソフトウェアアーキテクチャの振る舞い検証に関する考察, ソフトウェア工学の基礎 XVIー日本ソフトウェア科学会 FOSE2009, 査読有, 2009, pp. 267-274

[12] 沢田篤史, 小林隆志, 中道上, 他 3 名, 飛行船制御を題材としたプロジェクト型ソフトウェア開発実習, 情報処理学会論文誌, 査読有, 50 巻 11 号, 2009, pp. 2677-2689

〔学会発表〕(計9件)

- [1] 大森隆行、丸山勝久、林晋平、沢田篤史、ソフトウェア進化研究に関する動向調査 ～IWPSE シリーズを題材に～、電子情報通信学会ソフトウェアサイエンス研究会、2012年3月14日、てんぶす那覇
- [2] 張漢明、野呂昌満、沢田篤史、吉田敦、蜂巢吉成、横森励士、パターンに基づく CSP 記述の検査に関する考察、電子情報通信学会ソフトウェアサイエンス研究会、2011年10月28日、北陸先端科学技術大学院大学
- [3] 張漢明、野呂昌満、沢田篤史、蜂巢吉成、吉田敦、階層分割に基づく組込みソフトウェアの振舞い検証の支援について、情報処理学会第18回組込みシステム研究発表会、2010年8月10日、はこだて未来大学
- [4] 沢田篤史、On the Job Learning ～産学連携による新しいソフトウェア工学教育手法～、情報システム学会第3回シンポジウム、2010年6月26日、慶應義塾大学日吉キャンパス
- [5] 浦野彰彦、沢田篤史、野呂昌満、蜂巢吉成、デザインパターンを用いた CDI ツールのアーキテクチャとその実現、情報処理学会第168回ソフトウェア工学研究会/第17回組込みシステム研究会 合同研究発表会、2010年6月2日、化学会館
- [6] 加藤大地、沢田篤史、張漢明、野呂昌満、組込みソフトウェアの仕様モデルとアーキテクチャの関係に関する考察、情報処理学会研究報告第167回ソフトウェア工学研究会、2010年3月19日、国立情報学研究所
- [7] 張漢明、沢田篤史、野呂昌満、E-AoSAS++ における振舞い検証の枠組み、電子情報通信学会技術研究報告(ソフトウェアサイエンス研究会)、2010年3月8日、鹿児島大学
- [8] 蜂巢吉成、野呂昌満、沢田篤史、張漢明、遅延パーサを用いた軽量な XQuery 処理系、電子情報通信学会技術研究報告(ソフトウェアサイエンス研究会)、2010年3月8日、鹿児島大学
- [9] 吉田敦、大学の教育・研究活動のためのプログラム解析ツール、ウィンターワークショップ2010・イン・倉敷論文集(情報処理学会ソフトウェア工学研究会)、2010年1月21～22日、倉敷市芸文館

〔図書〕(計1件)

沢田篤史、平山雅之(編著)、三橋二彩子、丸山勝久、他6名(著)、CQ出版社、組込みソフトウェア開発技術、2011、352ページ

〔産業財産権〕

○出願状況(計0件)

○取得状況(計0件)

〔その他〕

6. 研究組織

(1) 研究代表者

野呂 昌満 (NORO MASAMI)
南山大学・情報理工学部・教授
研究者番号：40189452

(2) 研究分担者

沢田 篤史 (SAWADA ATSUSHI)
南山大学・情報理工学部・教授
研究者番号：40273841

研究分担者

蜂巢 吉成 (HACHISU YOSHINARI)
南山大学・情報理工学部・准教授
研究者番号：30319298

研究分担者

張 漢明 (CHANG HAN-MYUNG)
南山大学・情報理工学部・准教授
研究者番号：90329756

研究分担者

吉田 敦 (YOSHIDA ATSUSHI)
南山大学・情報理工学部・教授
研究者番号：50283495

(3) 連携研究者

横森 励士 (YOKOMORI REISHI)
南山大学・情報理工学部・准教授
研究者番号：40379152