様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　４ 年　 ６ 月 １３ 日現在

機関番号：３３９１７
研究種目：基盤研究(C)（一般）
研究期間：2019～2021
課題番号：１９Ｋ１１９１１

研究課題名（和文）アーキテクチャに基づく区間振る舞いモデルを用いた記述法と形式検証法

研究課題名（英文）Description method and formal verification method with section behavior model based on software architecture

研究代表者
張　漢明（Chang, Han-Myung）

南山大学・理工学部・准教授

研究者番号：９０３２９７５６

交付決定額（研究期間全体）：（直接経費）　　3,300,000円

研究成果の概要（和文）：IoTシステムでは複数のものが並列に動作するので，同時に起こる事象を考慮する必要がある．この事象の同時性を正しく認識できないことがシステム障害（failure）の原因となりうる．並列システムの正しさをテストだけを用いて保証することは難しい．システムの設計段階において，事象の同時性を含んだる舞い仕様の記述法とその検証法を体系化することにより，並列システムの設計品質の向上に寄与することが期待できる．

研究成果の学術的意義や社会的意義
並行システムの振る舞いの記述・検証のための言語としてプロセス代数では，事象の同時性を表現するためのステップ意味論などが理論的な研究は行われているが，同時性を考慮した振る舞い仕様の設計方法に関する研究成果はまだない．本研究では，事象の同時性を区間として局所化した構造を定式化して，その意味を既存のモデル検査を有するプロセス代数で定義することにより，事象の同時性を考慮した開発環境の基礎を築く．

研究成果の概要（英文）：Since multiple things operate in parallel in an IoT system, it is necessary to consider events that occur simultaneously. Failure to correctly recognize the simultaneity of these events can be the cause of system failures. It is difficult to guarantee the correctness of a parallel system by testing alone. Systematizing a method for describing and verifying event concurrency in the system design phase is expected to contribute to improving the quality of parallel system design.

研究分野：形式手法に基づくソフトウェア開発手法

キーワード：形式手法　形式仕様　形式仕様言語　モデル検査　ソフトウェアアーキテクチャ

１．研究開始当初の背景

全ての物にコンピュータが備えられ全ての物がネットワークで結びついた IoT システムでは，異なったシステムを統合してシステムのシステムを構築する必要がある．IoT システムは複数の実体が通信を介して動作する並列システムである．並列に動作するシステムでは，同時に発生する事象を考慮する必要がある．この並列事象の同時性を正しく認識できなければ，システム障害（failure）の原因となりうる．並列システムの振る舞いは，シリアライズ化された事象の系列を用いて分析が行われるので，通常，設計段階で事象の同時性は考慮されていない．事象の同時性に起因する障害の対処は，プログラミングの段階で行われることになる．プログラミングの段階におけるソフトウェアの欠陥修正は，設計の構造とプログラムの構造の不整合の要因となり，保守性の観点から問題となりうる．この問題は，並列システムにおける並列事象を並行システムとして実現するさいに，適切に並行事象として記述していないことに起因する．本研究では，並列事象の同時性を区間に局所化した「区間振る舞いモデル」を提案し，並列事象の同時性を並行システムとして実現するさいの振る舞い仕様の記述法とその検証法を体系化することにより，形式検証技術の実用化を目指した．

並行システムの信頼性を一般的なテスト技術を用いて保証することは困難であり，形式仕様言語を用いた仕様・設計の段階における検証が必要である．並行システムの振る舞いの記述・分析・検証のための言語として CSP や CCS に代表されるプロセス代数がある．伝統的なプロセス代数では，プロセスの並列合成の意味は非決定的な選択プロセスと同じなので，事象の同時性を表現することができない．文献[1]では，プロセス代数において事象の同時性を表現するためのステップ意味論や non-terminating serializable processes が紹介されている．ステップ意味論では，事象の多重集合を用いてインタリービング意味論による事象の同時性を表現可能にしている．non-terminating serializable processes はインタリービングの概念を用いずに，事象間の関係を記述することにより事象の同時性を表現している．事象の同時性が表現できるプロセス代数の言語とその意味論に関する研究は行われているが，同時性を考慮した振る舞い仕様の設計方法に関する研究成果はまだない．

並行システムでは一般的に同じ入力に対する振る舞いが非決定的なので，障害の原因となる欠陥（fault）の特定は難しい．並列システムでは，入力に対する振る舞いは非決定的かつ時間に依存するので，同時性を含んだ振る舞いの正しさをテストで検証することは本質的にできない．振る舞いに関する設計の正しさを，プログラムを実現する前の設計段階で検証することが重要である．設計段階において振る舞いを検証する技術としてモデル検査がある．モデル検査を用いて振る舞いを検証するためには，並列事象の同時性をどのように記述するかが課題となる．並列・分散システムでは同時に事象が発生することは本質的に避けられない．大域的な時間がない環境で，並列事象の同時生起を厳密に定義して，システムの網羅的な振る舞い仕様を簡潔に記述および検証する系統的な手法を提示する必要がある．

２．研究の目的

本研究の目的は，システムとシステムを統合する際に問題となる，並列システムにおける同時事象を考慮した振る舞い仕様を形式化し，その検証法を提示することにより，振る舞い仕様の形式的な検証を実用化することである．並列・分散システムにおいて，際どいタイミングで稀に発生する事象を従来のテスト技術で検証することは，事象発生の非決定性に起因して困難な作業である．際どいタイミングの問題を，並列システムの同時事象の問題として定式化することにより，振る舞い仕様に関して考慮すべき性質が明確になる．同時事象を区間としてカプセル化することにより，振る舞い仕様のモジュール化の標準化をはかる．

モデル検査は，自動で検査できるという観点から実用的なソフトウェア開発において検証の有用な手段として期待されている．モデル検査を実用化するためには，状態爆発を回避するための抽象化技術の開発が必要である．並列システムにおける同時事象を定義して，同時事象を区間に局所化することにより，同時事象の抽象化が可能になる．区間を単位とした振る舞いをこの抽象化された事象で記述することにより，システム全体の事象を削減することができる．事象の削減は，モデル検査における状態数の削減に寄与する．同時事象に関する性質は区間を単位としてカプセル化することにより，事象の並列性から事象の同時性を分離する．事象の同時性をモジュール化することにより，事象の同時性を考慮した振る舞い仕様の設計・検証が可能になる．

並列システムの同時事象を扱うための区間振る舞いモデルの考察は，形式手法を中心とした理論的な側面と，実際のソフトウェア開発事例による実践的な側面の両面から研究を進めることである．大域的な時計が存在しない非同期システムでは，事象が同時に発生することを，ある区間内で複数の事象が発生すること，と定義することができる．しかし，この区間を認識するさいに，必ず事象の伝搬に遅れが生じる．この遅れを考慮してシステム全体の振る舞いを設計する必要がある．このような詳細な仕様は段階的に行われるべきである．実践的な側面からは，仕様の表記法として，現在ソフトウェア開発で一般的に使われている UML 記法を導入する．形式記法と UML 記法を対応づけ相互変換を可能にすることにより本検証手法の実用性を高める．

本研究では，並列システムにおける複数事象の「同時性」の記述を並行プロセスとして記述することで，現実のソフトウェア開発において実用的な振る舞い記述法と検証法を提供することを目指す．区間の概念を導入することにより，考慮する必要のない並列事象の省略化と並列事象の抽象化による網羅的な振る舞い仕様の簡易化，および，振る舞い仕様と検証式のパターン化による再利用性の向上が期待できる．本研究では，既存の道具を利用することを想定している．UML 記法のサブセットを既存の形式仕様言語と対応づけることにより，仕様記述者は使い慣れた図式構文を用いてシステムの振る舞いを記述することができ，一方で，その意味は既存の形式仕様言語で理解することができる．汎用的な検証式をライブラリとして用意，もしくは，自動生成することにより，モデル検査の実用化に貢献することができる．

３．研究の方法

区間振る舞いモデルの定式化では，区間の構造の一般化と洗練化および簡易な構造の観点から，区間振る舞いモデルを再定義する．これまでに提案した区間振る舞いモデルから，区間を定義するために必要な概念を抽出し，それらの構造を数学モデルで表現する．これらの構造に対して形式仕様言語としてプロセス代数 CSP を用いて意味を定義する．CSP は，プロセス記述に関数プログラミングの機能を有する言語 CSP_M が提供されており，FDR などの実用的なモデル検査器が存在するので，振る舞いを表現するための形式言語として適切である．

区間振る舞いモデルの図式表現として，UML の振る舞い図の状態マシン図，アクティビティ図，コミュニケーション図などがある．区間振る舞いモデルとして並列性記述の観点からフォークとジョインを用いた状態マシン図もしくはアクティビティ図を用いることが考えられる．状態マシン図とアクティビティ図は，グラフ図式のノードを状態とみなすかアクションとみなすかの違いである．区間を状態とみなし状態を変更するきっかけを事象とすれば，状態マシン図が適切と思われる．区間を同時の事象を含む事象の列であるとみなせばアクティビティ図が適切であると思われる．区間振る舞いモデルでは，事象が区間内でおこりある条件で区間が変わるとみなせば，アクティビティ図が適切であると考えられる．区間振る舞いモデルの構造をアクティビティ図の構成要素に対応づけ，事例を用いて図式表現の有用性を検討する．

並行システムのデバッグ支援としてフォールトパターンを用いた欠陥の特定について考察す

る．プログラムのフォールトとプログラムを実行した軌跡の関係を定式化して，プログラムの動作の軌跡からプログラムのフォールトへの対応づけを「フォールトパターン」として提示する．フォールトと軌跡の間の関係は，フォールトの構造と軌跡の構造の間で対応づけられと想定する．「フォールト構造の集合」から「軌跡構造の集合」への写像を「誤り軌跡パターン」として特定できれば，その逆像である「軌跡構造の集合」から「フォールト構造の集合」への写像を「フォールトパターン」を得ることができる．並行プログラムにおける同期制御の基本構造は典型的な同期問題の解法にあると仮定して，同期問題の解法に対してプログラムの実行履歴から，プログラムのフォールトを特定する方法を検討する．

４．研究成果

本研究の成果として，(1)CSP を用いた区間振る舞いモデルの定式化，(2)UML を用いた区間振る舞いモデルの図式表現，(3)フォールトパターンによるデバッグ支援，について述べる．

(1) CSP を用いた区間振る舞いモデルの定式化

区間振る舞いモデルでは，ステップ意味論の記法を取り入れて同時に起こる事象を，集合を用いて表現する．区間は，区間の範囲を規定する開始境界と終了境界，入力事象集合と出力事象集合，および，入力事象集合と出力事象集合間の抽象化写像から構成される．区間は，基本区間を基にして，逐次区間，選択区間，および，並列区間の複合区間から構成される．自動販売機システムと現金自動預払機システムを事例として，区間振る舞いモデルを用いて記述した．

区間振る舞いモデルは，抽象度の高い検証可能な振る舞い表現を提供し，モデル検査の実用化に寄与する．区間の構造は，同時に起こる判断を区間に限定し，区間の構成に関わる事象だけに事象を限定する．区間の振る舞いはブラックボックス化して，区間内で同時事象を含めた起こり得る入力事象と出力事象を定義する．入力事象と出力事象の関係を，抽象化写像を用いて定義する．区間間の関係は，出力事象を用いて表現されるので，入力事象が未定義でも区間間の振る舞いを定義することができる．これが，モデル検査を実用化するための事象の数の削減に寄与する．入力事象と抽象化写像を用いて，詳細な振る舞いを表現することも可能である．

(2) UML を用いた区間振る舞いモデルの図式表現

対象システムを区間振る舞いモデルを用いて図式表現するために，区間の構成要素をアクティビティ図の構成要素に対応づけた．図１は，区間モデルのアクティビティ図の表現を示してい

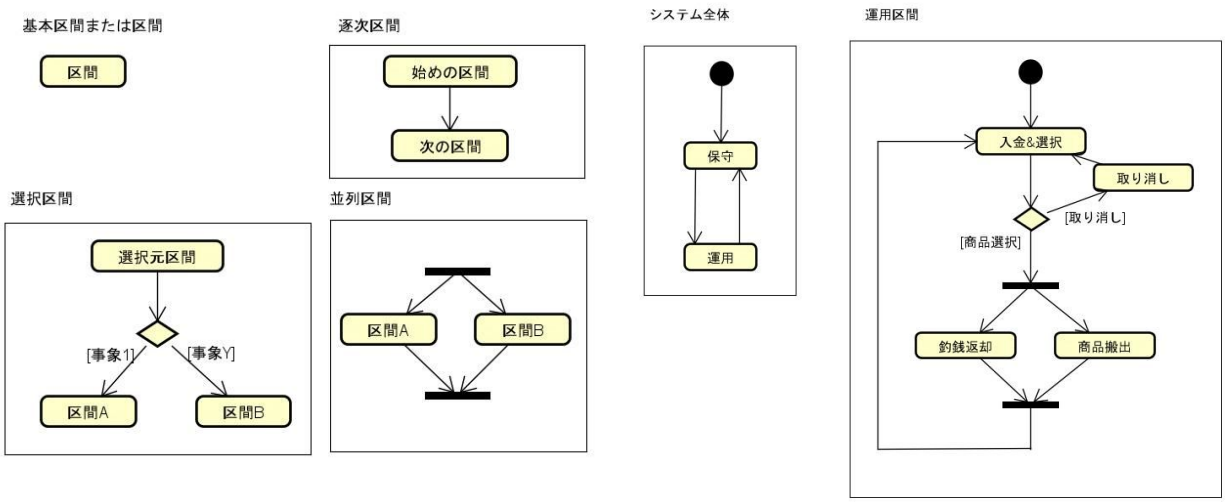

図 1 区間モデルアクティビティ図による表現

る．図の左側は，上記の区間の構成要素をアクティビティ図で表現したものである．図の右側は自動販売機の事例である．区間の変更は逐次区間および選択区間で行われる．事象の同時性はア

クション内に局所化されるので，これらの事象はアクション内で同時性を考慮した上で起こる事象とみなされる．自動販売機の事例では，「入金と選択」の区間で「取り消し」事象と「商品選択」事象により区間が変更される．これらの事象が起こるさいに，その他の事象との同時性は「入金と選択」区間で考慮されることになる．このようにして事象の同時性を区間内にカプセル化することにより状態遷移が簡潔になり，アクティビティ図を用いてシステム全体の振る舞いの見通しが良くなる．

(3) フォールトパターンによるデバッグ支援

並行プログラミングにおいて，システムの障害(failure)からプログラムのフォールト(fault)を特定するデバッグは困難な作業である．並行プログラムのデバッグでは，障害の状況から，障害に至る動作履歴を想定してプログラムのフォールトを特定する．並行プログラムではプロセスのコンテキストスイッチが非決定的に切り替わるので，検討すべき動作履歴は単一プロセスに比べて複雑になる．並行プログラムのデバッグは古くから扱われている問題であり，デバッグのための様々なツールやアプローチが継続的に提案されている．これら様々なデバッグ支援技術に共通する特徴は，対象プログラムの実行時に観測されるログなどの情報を扱っていることである．実行時の情報を，それぞれの観点からのモデルに基づいて抽象化し，フォールト箇所の特定や修正に用いている．

本研究では，並行プログラムにおける同期制御の基本構造は典型的な同期問題の解法にあると仮定して，同期問題の解法に対してプログラムの実行履歴から，プログラムのフォールトを特定する方法の提案を目指した．プログラムのフォールトとプログラム動作軌跡の間の関係が解明されれば，障害発生時の軌跡の情報からプログラムのフォールトへの対応付けが可能になると期待できる．

本研究の基本的なアイデアは，プログラムのフォールトと軌跡の関係を定式化して，プログラムの動作の軌跡からプログラムのフォールトへの対応づけを「フォールトパターン」として提示することである．フォールトと軌跡の間の関係は，フォールトの構造と軌跡の構造の間で対応づけられると想定した．「フォールト構造の集合」から「軌跡構造の集合」への写像を「誤り軌跡パターン」と定義できれば，その逆写像として「軌跡構造の集合」から「フォールト構造の集合」への写像を「フォールトパターン」として得ることができる．

本研究では，以下の方法で研究を進めた．（１）セマフォアを用いた同期問題の解法[2]を対象とする．（２）個々の具体的な問題に対してフォールトと軌跡の間の関係を定義する．(3) フォールトと軌跡の間の関係をパターン化する．具体的な問題を通して，フォールトの分類と軌跡の表現を洗練し，その後，フォールトの表現を構造化して抽象的な表現を検討した．フォールトを判別するための軌跡の表記法を提案し，第一種読み書き問題の解法に対する誤り軌跡集合を定義して，軌跡のパターンを発見すればフォールトを一意に特定可能であることを確認した．

参考文献

[1] Bergstra, J. A., Ponse, A., & Smolka, S. A., Handbook of Process Algebra, Elsevier, 2001.

[2] 土居範久，相互排除問題，岩波書店，2011.

５．主な発表論文等

〔雑誌論文〕　計5件（うち査読付論文　5件／うち国際共著　0件／うちオープンアクセス　2件）

１．著者名 江坂篤侍，野呂昌満，繁田雅信，沢田篤史	４．巻 26
２．論文標題 ソフトウェアアーキテクチャに基づく組込みシステムの設計法に関する研究	５．発行年 2019年
３．雑誌名 ソフトウェア工学の基礎XXVI（日本ソフトウェア科学会FOSE2019）	６．最初と最後の頁 pp．151-156
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 －

１．著者名 横山史明，沢田篤史，野呂昌満，江坂篤侍	４．巻 26
２．論文標題 IoTの柔軟な相互運用性を実現するソフトウェアアーキテクチャの提案	５．発行年 2019年
３．雑誌名 ソフトウェア工学の基礎XXVI（日本ソフトウェア科学会FOSE2019）	６．最初と最後の頁 pp．93-102
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 －

１．著者名 横山史明，沢田篤史，野呂昌満，江坂篤侍	４．巻 62
２．論文標題 IoTの柔軟な相互運用性を実現するソフトウェアアーキテクチャの提案	５．発行年 2021年
３．雑誌名 情報処理学会論文誌	６．最初と最後の頁 pp．995-1007
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

１．著者名 野呂昌満，沢田篤史，張漢明，繁田雅信	４．巻 1
２．論文標題 アスペクト指向アーキテクチャに基づく組込みソフトウェアの設計法の提案	５．発行年 2021年
３．雑誌名 ソフオツェアエンジアリングシンポジウム2021論文集	６．最初と最後の頁 pp．32-40
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

1．著者名 Akira Mizutani, Masami Noro, Atsushi Sawada	4．巻 1
2．論文標題 Design of Software Architecture for Neural Network Cooperation: Case of Forgery Detection	5．発行年 2021年
3．雑誌名 Proceedings of 2021 28th Asia-Pacific Software Engineering Converence	6．最初と最後の頁 pp. 130-140
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

〔学会発表〕　計5件（うち招待講演　0件／うち国際学会　1件）

1．発表者名 Tomohiro Oda, Keijiro Araki, Yasuhiro Yamamoto , Kumiyo Nakakoji, Han-Myung Chang, and Peter Gorm Larsen
2．発表標題 Specifying Abstract User Interface in VDM-SL
3．学会等名 THE 19TH OVERTURE WORKSHOP（国際学会）
4．発表年 2020年

1．発表者名 張漢明，野呂昌満，沢田篤史
2．発表標題 同時に起こる事象を考慮した区間振る舞いモデルの提案
3．学会等名 第56回組込みシステム研究発表会
4．発表年 2021年

1．発表者名 張漢明，野呂昌満，沢田篤史
2．発表標題 MVCに基づいた組み込みソフトウェアの形式仕様メタモデルに関する考察
3．学会等名 情報処理学会第53回組込みシステム研究会
4．発表年 2020年

１．発表者名
小澤司、青山幹雄、沢田篤史，野呂昌満

２．発表標題
業務の依存関係分析に基づくWebシステムアーキテクチャの再設計方法に関する研究

３．学会等名
電子情報通信学会ソフトウェアサイエンス研究会（SS2021-15）

４．発表年
2021年

１．発表者名
張漢明，高木裕也，沢田篤史，野呂昌満

２．発表標題
並行システムデバッグ支援のためのフォールトパターンに関する考察

３．学会等名
情報処理学会第59回組込みシステム研究会

４．発表年
2022年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

―

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考
研究分担者	野呂　昌満 (Noro Masami) (40189452)	南山大学・理工学部・教授 (33917)	
研究分担者	沢田　篤史 (Sawada Atsushi) (40273841)	南山大学・理工学部・教授 (33917)	

７．科研費を使用して開催した国際研究集会

〔国際研究集会〕　計0件

８．本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関