

令和 4 年 6 月 6 日現在

機関番号：32675  
研究種目：基盤研究(B)（一般）  
研究期間：2017～2020  
課題番号：17H01726  
研究課題名（和文）リアクティブプログラミングのための高信頼化技術

研究課題名（英文）Technology for Reliable Reactive Programming

研究代表者

細部 博史（Hosobe, Hiroshi）

法政大学・情報科学部・教授

研究者番号：60321577

交付決定額（研究期間全体）：（直接経費） 14,460,000円

研究成果の概要（和文）：リアクティブプログラミングの高信頼化技術の構築を目的として、リアクティブシステムの分析と処理に関する研究を行った。その成果として、視覚的・対話的なプログラムを対象としたイベント駆動プログラミング言語によるイベント処理に関するユニットテスト手法、情報可視化を対象としたデータ駆動文書のためのアプリケーションプログラミングインタフェースの改良手法、因果ループ図の形式定義と性質解析の手法、関数リアクティブプログラミングにおけるグリッチの回避の手法を提案した。

研究成果の学術的意義や社会的意義

本研究成果の学術的意義は、現代のソフトウェアの中心であるリアクティブシステムに関して、その一形態であるイベント駆動型の視覚的・対話的なプログラムを対象として、その適切な振舞いを記述し、テスト時にプログラムの不具合を検出する手法を構築したことである。本研究成果の社会的意義は、ソフトウェアの高信頼化という問題に対して、イベント駆動型プログラムのテスト手法の観点から新たな可能性を示したことである。

研究成果の概要（英文）：To construct technology for reliable reactive programming, we studied analysis and processing of reactive systems. Our research results include a method of unit testing for processing events in an event-driven programming language for visual and interactive programs, a method for improving the application programming interface for data-driven documents for information visualization, a method for formal definition and analysis of causal loop diagrams, and a method for avoiding glitches in functional reactive programming.

研究分野：プログラミング

キーワード：ソフトウェア 高信頼化 リアクティブプログラミング

## 1. 研究開始当初の背景

1985年にHarelとPnueli(引用文献①)は、外界から繰り返し刺激され、外的入力に対して連続的に反応し、環境と継続的な関係を維持するシステムをリアクティブシステムと名付けた。イベント駆動や対話的なアプリケーションに代表される現代のソフトウェアの多くは、リアクティブシステムである。しかし、リアクティブシステムの適切な実現は容易ではない。例えばAdobeの巨大なコードベースでは約1/3のコードがイベント処理や他のGUIコンポーネントのために費やされていたが、報告されたバグの半数以上がそれらに起因するものであった(引用文献②)。このことは、従来のイベント駆動アプリケーションにおける、いわゆるコールバックを用いたイベント処理に限界があることを示唆している。このような背景のもと、リアクティブシステムに対するソフトウェア開発者の意識は高まっている。事実、J. Bonérらによるリアクティブシステムの重要性に関する宣言“The Reactive Manifesto”(引用文献③)は2016年10月時点で16,000名以上の署名を集めている。

リアクティブシステムの開発に対するアプローチの一つとして、リアクティブプログラミング(引用文献④)が注目されている。リアクティブプログラミングは、外的なイベントに対する反応を抽象化して扱い、データと計算の依存関係を自動管理する機構を提供する。これは表計算ソフトウェアで、あるセルの値が変化すると、フォーミュラを通じて他のセルの値が自動更新される機構に類似している。以前にも制約プログラミングやデータフロープログラミングで類似の概念が実現されていたが、これらのプログラミング方式は研究の広がりや一般の開発者への普及という点では十分に成功したとは言い難かった。

他方、リアクティブプログラミングは近年、様々な広がりを見せている。Fran(引用文献⑤)は最初期のシステムであり、関数プログラミング言語Haskell上に対話的なアニメーションを構成するためのライブラリである。Fran以降、いくつものシステムが関数プログラミング言語、あるいは関数プログラミング的な環境上に構築されており、関数型リアクティブプログラミングと呼ばれている。イベントはストリームとして見なされ、イベントに対する振舞いは関数として記述される。これによって、関数プログラミングに適した形で、イベント駆動アプリケーションを記述できるようになる。GUIの構築などは以前の制約プログラミングでも扱われていたが、リアクティブプログラミングではより現代的なウェブも主要な応用とされている。例えばFlapjax(引用文献⑥)はJavaScript上に構築された言語で、Ajaxによるウェブアプリケーションを構築できる。さらにリアクティブプログラミングは研究領域に留まらず、企業等の一般の開発にも浸透しつつある。例えばMicrosoftは.NET向けのライブラリRxを開発し、JavaScriptやC++等の他言語にも移植している。Facebookはウェブアプリケーションを構築するためのJavaScriptライブラリReact.jsを開発している。

このようにリアクティブプログラミングは普及の兆しを見せているが、その技術面では十分に成熟しているわけではない。例えばグリッチと呼ばれる、意図しない振舞いを引き起こす問題があることが知られている(引用文献④)。グリッチとは、あるイベントによる影響が複数の経路を通じて同一の部品に到達した際に、その遅延の差のために瞬間的に誤った出力を生じる問題である。グリッチはスタンドアロンのシステムでは比較的容易に解決可能であるが、分散システムでは十分な解決がなされていない。また、リアクティブプログラミングではデバッグが困難である(引用文献⑦)。本質的にイベント駆動であり、イベント処理のためのプログラムの実行が複数の部品間にまたがるため、従来の命令型プログラミングで広く用いられている、ステップ実行やブレークポイント等によるデバッグ方式が適さないためである。今後、イベント処理に対する多方向性(引用文献④)の導入などの拡張も検討されているが、このような新たな拡張への対応も含めて、高信頼なリアクティブプログラミングを実現するための技術の構築が必要である。

## 2. 研究の目的

本研究は、リアクティブプログラミングの高信頼化技術の構築を目的とする。特にリアクティブプログラミングによって複数の部品が合成されたリアクティブシステムの分析と処理に焦点を合わせ、三つのアプローチからこの課題に取り組む。

## 3. 研究の方法

制約プログラミング、ゲーム理論、アダプティブ計算の三つのアプローチを採用する。より具体的には、制約プログラミングに基づくリアクティブシステムの不具合箇所発見、ゲーム理論に基づくリアクティブシステムの振舞いの分析、アダプティブ計算に基づくリアクティブシステムの安全かつ効率的な処理に関する研究を行う。

## 4. 研究成果

### (1) イベント駆動プログラミング言語のためのユニットテスト手法

リアクティブシステムの一形態であるイベント駆動型の視覚的・対話的なプログラムを対象として、イベント処理に関するユニットテストの手法を構築し、イベント駆動プログラミング言

語である Processing に適用して本手法の処理系を開発した。本処理系を用いることで、視覚的・対話的なプログラムの適切な振舞いを記述し、テスト時にプログラムの不具合を検出することができる。

本研究で提案した手法は、テスト可能な Processing プログラムを Java で記述できる。ただし、Java による Processing プログラムの記述方法では通常、メインプログラムをクラス PApplet のサブクラスとして記述するのに対して、本手法ではメインプログラムをクラス PTestableApplet のサブクラスとして定義する。また、本手法では、Java と JUnit でテストプログラムを記述する。他の多くのユニットテストツールと同様、類似のテストケースをまとめるテストメソッドを指定し、テストケースはアサーションによって指定する。本手法の実現のため、PTestableApplet は二つのモードを導入している。ノーマルモードは、単純に PApplet のメソッドを呼び出すことで、通常の Processing プログラムと同様に動作する。一方、テストモードは、テストプログラムに指定されたイベントを生成し、対象とする Processing プログラムに定義されたイベントハンドラを呼び出すことで、Processing の実行をシミュレートする。

本手法の有効性を評価するために、イベント駆動型の視覚的・対話的なプログラムを Processing で作成し、そのユニットテストに関する実験を行った。その結果、マウスで操作する GUI 部品の振舞いや、ビデオゲームのようにキーボードで操作する視覚的オブジェクトの振舞いに関するプログラムの不具合検出に、本手法が適用可能であることが示された。

#### (2) データ駆動文書のためのアプリケーションプログラミングインタフェースの改良手法

データ駆動文書と呼ばれる情報可視化のためのプログラミング方式に関して、そのアプリケーションプログラミングインタフェース (API) を改良する手法を提案した。これによって、様々な可視化に対応できるように API の一般性を保ちつつ、データ駆動文書による情報可視化のためのプログラムの記述性、可読性、信頼性を向上している。

本研究では三つの方法を採用した。まず、TypeScript 言語による静的型付けの利用によって、属性名と属性値の型を静的に検査できるようにした。次に、メソッドチェーンの削減によって、一つのメソッドで複数の属性に値を設定できるようにした。さらに、仮想属性の概念の導入によって、text、style 等の構成要素を属性と同様に扱えるようにした。

#### (3) 因果ループ図の形式定義と性質解析

上位のモデリングからリアクティブプログラミングまでの滑らかな開発法の確立について検討した。本研究では、因果ループ図の操作的な意味に形式定義を与え、振舞い仕様の性質を自動検査する方法を提案し、ネット指向形式体系の採用によって、因果ループ図の特徴である真の並行性の表現を可能にした。

#### (4) 関数リアクティブプログラミングにおけるグリッチの回避

関数リアクティブプログラミングにおける循環データフローの除去によるグリッチの回避の手法を構築した。本手法は、再帰関数の組化や末尾再帰化などの最適化により、循環したデータフローに対応する再帰計算が反復計算に帰着可能ならば、データフローの循環を除去することができる。循環を除去できれば、出力の差分を入力差分から直接計算することができるため、データフローの循環によるグリッチを避けることができる。また、循環のないデータフローに対応する計算は、可逆計算とその漸進化の枠組により、出力の差分を入力差分に置き換えることができ、リアクティブ計算の多方向化を実現できる。

#### <引用文献>

- ① D. Harel and A. Pnueli: On the Development of Reactive Systems, Logics and Models of Concurrent Systems, NATO ASI Series, vol. 13, pp. 477-498, 1985.
- ② J. Järvi, et al.: Property Models: From Incidental Algorithms to Reusable Components, Proc. GPCE, pp. 89-98, ACM Press, 2008.
- ③ J. Bonér, et al.: The Reactive Manifesto, v2.0, 2014.  
<http://www.reactivemanifesto.org/>
- ④ E. Bainomugisha, et al.: A Survey on Reactive Programming, ACM Comput. Surv., 45(4):52:1-34, 2013.
- ⑤ C. Elliott and P. Hudak: Functional Reactive Animation, Proc. ICFP, pp. 263-273, ACM Press, 1997.
- ⑥ L. A. Meyerovich, et al.: Flapjax: A Programming Language for Ajax Applications, Proc. ACM OOPSLA, pp. 1-20 2009.
- ⑦ G. Salvaneschi and M. Mezini: Debugging for Reactive Programming, Proc. ICSE, pp. 796-807, ACM Press, 2016.

## 5. 主な発表論文等

〔雑誌論文〕 計6件（うち査読付論文 5件/うち国際共著 1件/うちオープンアクセス 0件）

1. 著者名 Hironari Kawada, Mina Akaishi, and Hiroshi Hosobe	4. 巻 -
2. 論文標題 A Graph-Based Visualization of Time-Series Information in Multiple Texts	5. 発行年 2018年
3. 雑誌名 Proceedings of the 22nd International Conference on Information Visualisation (IV2018)	6. 最初と最後の頁 44-49
掲載論文のDOI (デジタルオブジェクト識別子) 10.1109/iv.2018.00018	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 Yota Inayama and Hiroshi Hosobe	4. 巻 -
2. 論文標題 Toward an Efficient User Interface for Block-Based Visual Programming	5. 発行年 2018年
3. 雑誌名 Proceedings of the 2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC2018)	6. 最初と最後の頁 293-294
掲載論文のDOI (デジタルオブジェクト識別子) 10.1109/VLHCC.2018.8506530	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 Guillermina Cledou and Shin Nakajima	4. 巻 -
2. 論文標題 A Net-based Formal Framework for Causal Loop Diagrams	5. 発行年 2018年
3. 雑誌名 Proceedings of the 3rd Asia-Pacific Conference on Complex Systems Design and Management (CSD&M Asia 2018)	6. 最初と最後の頁 1-12
掲載論文のDOI (デジタルオブジェクト識別子) 10.1007/978-3-030-02886-2_1	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 該当する
1. 著者名 Hiroshi Hosobe	4. 巻 -
2. 論文標題 Testing Event-Driven Programs in Processing	5. 発行年 2020年
3. 雑誌名 Proceedings of the 2020 European Symposium on Software Engineering (ESSE2020)	6. 最初と最後の頁 6-11
掲載論文のDOI (デジタルオブジェクト識別子) 10.1145/3393822.3432338	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 Soichiro Hidaka	4. 巻 -
2. 論文標題 Trace-based Error Prevention and Detection of Consistency Maintenance via Bidirectional Graph Transformations	5. 発行年 2020年
3. 雑誌名 Companion Proceedings of the 20th IEEE International Conference on Software Quality, Reliability and Security (QRS2020)	6. 最初と最後の頁 87-90
掲載論文のDOI (デジタルオブジェクト識別子) 10.1109/QRS-C51114.2020.00025	査読の有無 無
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 Ami Sakakibara and Hiroshi Hosobe	4. 巻 12785
2. 論文標題 A Video Game-Like Approach to Supporting Novices in Learning Programming	5. 発行年 2021年
3. 雑誌名 Lecture Notes in Computer Science (HCI International 2021)	6. 最初と最後の頁 84-93
掲載論文のDOI (デジタルオブジェクト識別子) 10.1007/978-3-030-77943-6_6	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10件 (うち招待講演 1件 / うち国際学会 2件)

1. 発表者名 中島震, Guillermina Cledou
2. 発表標題 因果ループ図の形式定義と性質解析
3. 学会等名 情報処理学会第199回ソフトウェア工学研究発表会
4. 発表年 2018年

1. 発表者名 Shin Nakajima
2. 発表標題 Formal Analysis of Causal Loop Diagrams
3. 学会等名 Asia Pacific Conference on Robot IoT System Development and Platform (APRIS2018) (招待講演) (国際学会)
4. 発表年 2018年

1. 発表者名 飯岡徹人, 若原徹, 細部博史
2. 発表標題 変換画像の同時学習によるICMの改良
3. 学会等名 人工知能学会第109回人工知能基本問題研究会
4. 発表年 2019年

1. 発表者名 河田裕成, 細部博史, 赤石美奈
2. 発表標題 有向グラフとCoordinated Multiple Viewsによる時系列情報の可視化
3. 学会等名 情報処理学会第81回全国大会
4. 発表年 2019年

1. 発表者名 相原崇弘, 日高宗一郎
2. 発表標題 双方向変換を用いたコードクローン間の選択的な更新伝播手法
3. 学会等名 情報処理学会第123回プログラミング研究会
4. 発表年 2019年

1. 発表者名 細部博史
2. 発表標題 Processingプログラムにおけるイベント処理のユニットテスト手法
3. 学会等名 第22回プログラミングおよびプログラミング言語ワークショップ(PPL2020)
4. 発表年 2020年

1. 発表者名 日高宗一郎
2. 発表標題 トレースに基づく双方向変換の多方向化に向けたco-targetial compositionのための漸進化
3. 学会等名 第22回プログラミングおよびプログラミング言語ワークショップ(PPL2020)
4. 発表年 2020年

1. 発表者名 細部博史
2. 発表標題 イベント駆動に基づくProcessingプログラムのユニットテスト手法
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(2020-03-SS)
4. 発表年 2020年

1. 発表者名 Hiroshi Hosobe
2. 発表標題 A Tool for Testing Event-Driven Programs in Processing
3. 学会等名 3rd International Workshop on User Interface Test Automation and Testing Techniques for Event Based Software (INTUITESTBEDS2020) (国際学会)
4. 発表年 2020年

1. 発表者名 細部博史
2. 発表標題 データ駆動文書のためのAPIの改良
3. 学会等名 第24回プログラミングおよびプログラミング言語ワークショップ(PPL2022)
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	中島 震  (Nakajima Shin)  (60350211)	国立情報学研究所・大学共同利用機関等の部局等・名誉教授   (62615)	
研究 分担者	日高 宗一郎  (Hidaka Soichiro)  (70321578)	法政大学・情報科学部・教授   (32675)	

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
---------	---------