

令和 4 年 8 月 26 日現在

機関番号：12608

研究種目：基盤研究(C) (一般)

研究期間：2018～2021

課題番号：18K11236

研究課題名(和文) アクターモデルと関数リアクティブプログラミングの融合による組込みシステム開発支援

研究課題名(英文) Embedded Systems Development Support with the Integration of the Actor-Model and Functional Reactive Programming

研究代表者

渡部 卓雄 (Watanabe, Takuo)

東京工業大学・情報理工学院・教授

研究者番号：20222408

交付決定額(研究期間全体)：(直接経費) 3,400,000円

研究成果の概要(和文)：アクターモデルの考え方を取り入れた関数リアクティブプログラミング(FRP)言語が、計算資源の限られた組込みシステムの安全性と開発効率向上に寄与することを以下の観点で示した。(1)分散システムにおいてグリッチを回避しつつ効率的に時変値の変化を伝播させるアルゴリズムと、それにもとづく分散FRP言語を提案した。(2)状態遷移による振る舞いの変化の記述を支援する機構をFRP言語に導入し、それによる状態依存動作のモジュール性向上を示した。(3)データの最大サイズを伴う再帰型を定義できる型システムをもつFRP言語を提案し、型付けされたプログラムは規定されたメモリ量で安全に実行できることを証明した。

研究成果の学術的意義や社会的意義

アクターモデルを実行系とする分散関数リアクティブプログラミング(FRP)言語は、関数プログラミングが提供する諸性質(高い可読性と実行効率、形式的扱いの容易さ等)をネットワークで相互接続された組込みシステムの領域に導入するものであり、国内外を通してユニークである。また、状態依存動作の抽象化機構、およびプログラムの使用メモリ量をコンパイル時に保証できる型システムは、いずれも組込みシステムの開発効率と安全性の向上に寄与する。本研究の成果は、IoTやサイバーフィジカルシステムを含む広範な組込みシステムの信頼性と安全性および開発効率の向上に貢献することが期待できる。

研究成果の概要(英文)：This study shows that using the Functional Reactive Programming (FRP) language, which incorporates the idea of actor models, contributes to improving embedded systems' safety, reliability, and development efficiency with limited computational resources. The main contributions are: (1) We proposed an efficient distributed glitch-free change-propagation algorithm for time-varying values. We designed and implemented an actor-based distributed FRP language that adopts the algorithm. (2) We introduced an explicit state description mechanism into an FRP language for small-scale embedded systems. The mechanism can improve the modularity of components with state-dependent behaviors. (3) We presented a novel type system that can define recursive data types with the maximum data size to be constructed. We designed and implemented an FRP language incorporating the type system. We proved that any well-typed program can safely be executed within the amount of the memory specified by the types.

研究分野：プログラミング言語

キーワード：関数リアクティブプログラミング 関数プログラミング言語 組込みシステム 分散システム アクターモデル 型システム

科研費による研究は、研究者の自覚と責任において実施するものです。そのため、研究の実施や研究成果の公表等については、国の要請等に基づくものではなく、その研究成果に関する見解や責任は、研究者個人に帰属します。

1. 研究開始当初の背景

現在大小様々な組込みシステムが用いられており、その信頼性と安全性および開発効率の向上は計算機科学における重要な課題の一つである。組込みシステムのソフトウェア開発を複雑にする要因の一つが、組込みシステムがリアクティブシステムであることである。

リアクティブシステムとは、離散的なイベントや外界の連続的な変化などの入力に対して、応答の生成および自身の状態の更新をし続けるシステムをいう。組込みシステムや GUI 等の対話的システムはリアクティブシステムとして表現できる。一般にリアクティブシステムに対する入力は、与えられる順序やタイミングはあらかじめ予測できない。そのため、リアクティブシステムのプログラミングでは、ポーリングやコールバックといった手法がスレッドや状態機械と共に用いられる。これらは一般的な手続き型プログラミング言語で実現可能である一方、コードの細分化を招き可読性およびモジュール性低下の原因となり、結果として信頼性と安全性および開発効率向上の妨げとなる。

関数リアクティブプログラミング (*Functional Reactive Programming, FRP*) は、離散的なイベント列を表すイベントストリームや時間と共に連続的に変化する値を表す時変値 (*time-varying value*) といった抽象化機構を関数プログラミングに導入することで、リアクティブシステムの動作を副作用を用いず宣言的に記述することを可能にする。

FRP 言語によるプログラム例を以下に示す。

```
fan = abs(tmp1 - tmp2) >= 5
```

このわずか 1 行のプログラムは、天井と床にそれぞれ 1 個ずつ設置した温度センサの計測値にもとづいてファンを制御しつづけるものである。計測値の差が 5 度以上のときにファンを ON にして部屋の空気を攪拌し、それ以外のときはファンを OFF にする。この例では温度センサの計測値を表す tmp1 と tmp2、およびファンの状態を表す fan が時変値である。また abs は絶対値関数である。

従来の書き方では、センサの値を繰り返し読む (ポーリング) か、センサからの割り込みによって計測値の変化を知り (コールバック)、ファンの状態を更新するよう記述する必要がある。一方 FRP 言語の場合、この例のように時変値間の関連を記述するだけで、入力の変化にしたがって出力が自動的に変化するようなプログラムを簡潔に書くことができる。

FRP は関数プログラミング言語 Haskell による (副作用に依らない) 対話的プログラムの記述手法として 1990 年代に考案され現在も発展している。また FRP の考え方は純粋な関数型言語以外においても有用であることが明らかになり、現在、Java, Scala, Javascript 等の言語で動作する RP ライブラリや DSL が GUI や Web アプリケーションの開発に用いられている。その後 2000 年代になって組込みシステムやロボットの記述にも有用であることが明らかになったが、これに関しては以下のような課題が挙げられる。(a) **計算資源の制約**: FRP の導入によって恩恵を受けるのはマイクロコントローラで構成されるような単機能・小規模な組込みシステムであり、Haskell などの実行系は現時点ではこれらで動作させるのは現実的ではない。(b) **十分な表現力**: 例えば非同期処理、分散処理、例外処理等、FRP によって必ずしも素直に表現できるとは限らない処理がある。(c) **変化する実行環境への適応**: 組込みシステムは実行環境の変化に対応しなければならないケースが多いが、前項 (b) と同様にそれを FRP で表現する手法は確立されていない。(d) **テストおよび形式的な検証**: FRP は本来純粋な関数型言語でリアクティブな動作を記述するためのものであり、動作を形式的に記述することは可能である。しかし現在のところ FRP で記述されたプログラムのテストおよび形式的な検証手法は確立されているとは言えない。

2. 研究の目的

アクターモデルを実行系に用いた自己反映的関数リアクティブプログラミング (FRP) 言語を設計・実装し、それが組込みシステムの信頼性と安全性および開発効率の向上に寄与することを明らかにする。具体的には、当該言語を用いることで (A) 計算資源に制約がある小規模組込みシステムにおいても効率的な実行が可能であること、(B) 十分な表現力とモジュール性を提供できること、(C) 変化する実行環境への適応的な動作が記述できること、および (D) テストおよび形式的な検証が可能であることをそれぞれ示す。

3. 研究の方法

(1) アクターモデルにもとづく分散 FRP 言語

リアクティブプログラミングの考え方は分散システムにおいても有用であり、現在までにいくつかの分散システム向けのリアクティブプログラミング言語やフレームワーク等が提案されている。リアクティブプログラミングを実現する上での問題としてグリッチ (glitch) が挙げられる。ここでグリッチとは、時変値の変化が複数の経路を通して伝播した場合に、伝播の時間差によって本来起こり得ない状態が発生することを指す。特に分散システムにおける通信の遅延はグリッチの回避を困難にする。

本研究では、アクターモデル上でグリッチを回避しつつ効率的に時変値の変化を伝播させるアルゴリズムと、それにもとづく分散 FRP 言語の実現手法を提案する。提案するアルゴリズムの正しさを示すとともに、提案手法にもとづく言語処理系を実装し、その有用性を評価する。

(2) 状態依存動作のための抽象化機構

FRP 言語 Emfrp は、マイクロコントローラのような限られた計算資源をもつ小規模組込み環境のためのソフトウェア開発を支援するために、研究代表者の研究室で設計・開発された。Emfrp はリソースの制約された環境で安全に動作することを目指して設計されているため、高階の時変値等、時変値の依存関係を動的に変化させるような言語機構は存在していない。一方組込みシステムの設計において、システムの動作を状態遷移系として定式化することは有用であり、モデルベース開発 (MBD) 等、状態遷移系としての記述をベースとする開発手法は広く行われている。このとき変化する状態に応じてシステムの振る舞いを変化させるようなことも一般的であるが、Emfrp では時変値間の依存関係が固定されているため、状態の変化に応じた振る舞いの変化を自然な形で記述するのは難しい。

この問題を解決するため、状態に依存して振る舞いに変化するコンポーネントの記述に特化した言語機構を提案する。これにより、入力に対する応答を宣言的に記述できる FRP の利点を享受しつつ直感的に状態遷移を扱うことができるようになる。また、システムが取りうる状態はプログラム中で定義したものに限定されるため、実行時に切り替える計算の集合をコンパイル時に決定することができる。したがって、静的にメモリの使用量を見積もることができる Emfrp の特徴を保ったまま動的な計算の切り替えを実現できるようになる。

(3) FRP 言語への再帰データ型の導入

Emfrp では実行時に使用するメモリ量が静的に決定されるため、動的なメモリ確保を必要としない。加えて時変値の更新処理の停止性が保証されているため、プログラム実行中にメモリリソースが枯渇することなく安全にリアクティブな動作を続けることができる。しかしこのような安全性を保証するために、Emfrp では時変値を第一級として扱わない、高階関数の定義使用を許さない、データ型や関数の再帰的定義使用を許さないといった言語設計上の制限を導入している。特にデータ型や関数の再帰的定義使用の禁止により、リストや木構造のような有用なデータ構造を自然な表現で扱うことが難しい。

本研究では、Emfrp の持つ性質を維持しながらも、再帰的定義に関する制限を緩和するために「構築できる構造の最大サイズ」を伴った再帰データ型で Emfrp を拡張した言語 $\text{Emfrp}^{\text{BCT}}$ を提案する。 $\text{Emfrp}^{\text{BCT}}$ では再帰データ型と同時に、関数呼び出しの際の引数のサイズが減少することを保証した再帰関数 (原始帰納的関数) を導入する。ここで導入するデータ型とそれに付随する言語機構は、先行研究である Sized Types を Emfrp のプログラミングスタイル (直前値を積極的に活用するスタイル) に合わせて拡張したものになっている。

提案した言語 $\text{Emfrp}^{\text{BCT}}$ について、(a) 型検査の際にサイズに関する検査が同時に行われるため、メモリを無制限に使用するプログラムがコンパイル・実行されることはないこと、および (b) ノード更新の停止性が保証されるため、実行時にメモリリソースが枯渇せずにリアクティブな動作を継続できることそれぞれの形式的な証明を与える。加えて、従来の Emfrp ではデータを固定サイズのデータ (タプル等) によって管理せざるを得なかったプログラムが、 $\text{Emfrp}^{\text{BCT}}$ ではリストやヒープ木を用いて簡潔かつ保守性に富んだ記述で実現できることを例題を通して示す。

4. 研究成果

(1) アクターモデルにもとづく分散 FRP 言語

非同期メッセージにもとづく時変値の変化伝播アルゴリズムを提案し、同アルゴリズムにもとづいたアクターモデルによる実行系をもつ分散 FRP 言語 Distributed-XFRP を設計・実装した。同言語を用いたいくつかの例題の実装とその評価を通して、提案手法の有用性を明らかにした。

他の時変値に依存しない時変値をソース (source) と呼ぶ。例えばセンサーの計測値を表す時変値が相当する。提案したアルゴリズムでは、ソースとなる時変値それぞれについて、それに依存する時変値間でのグリッチが生じないこと (single-source glitch freedom) が保証される。一方、2 個以上のソースの組に依存する時変値が複数個ある場合、それらの間では組を構成するソースの変化のタイミングが異なって観測されることは許容する (多くのアプリケーションにおいて実用上問題はない)。

Distributed-XFRP は、このアルゴリズムにもとづいた時変値の伝播機構をアクターモデル上の計算として実現した実行系を持つ。具体的には、Distributed-XFRP のコードは (アクターモデルにもとづく言語) Erlang のコードに翻訳され、Erlang が動作する環境で実行される。学会発表 [3] において、例題を通して Distributed-XFRP の評価 (実行効率および記述性) を行った。

また、Distributed-XFRP による無線センサーアクターネットワーク (WSAN) の記述方式についての提案を行った。この提案では、センサーノードとアクターノード (センサーの計測値を集約したり、アクチュエーターを動作させる役割を持つ) が混在するネットワークを単一の FRP のプログラムとして簡潔に記述できることを示している。

(2) 状態依存動作のための抽象化機構

小規模組込みシステム向け FRP 言語 Emfrp に対して、状態遷移による振る舞いの変化の記述に特化した言語機構 (switch 拡張) を導入した言語 XStorm を設計・実装し、同機構による状態依存動作を含むコンポーネントのモジュール性の向上を明らかにした。switch 拡張は、状態ごとに時変値と状態遷移の定義を行う機構を提供することで、FRP によるリアクティブシステムの宣言的な記述とステートマシンによる状態遷移の直感的な管理を両立させている。また、各状態の定義の独立性を高めることで可読性・拡張性の高いコードを記述することを可能にしている。さらに、switch 拡張は (拡張を導入する前の) Emfrp への変換として実装されているため、プログラムが実行時に使用するメモリ量が静的に決定できることなどの、小規模組込みシステムへの利用に有利な性質は保たれている。

(3) FRP 言語への再帰データ型の導入

構築できる構造の最大サイズを伴った再帰データ型を定義できる型システム Bounded-Construction-Types (BCT), およびそれを Emfrp に導入した言語 Emfrp^{BCT} を提案した。具体的な貢献は以下の通りである。

- Emfrp^{BCT} の構文, 操作的意味論, 型システムについて形式的定義を与えたのち, 式の評価に必要なメモリ量を決定するアルゴリズムを提示した。
- Emfrp^{BCT} の定義を用いて, 適切に型付けされサイズの妥当性の検査に通ったノードについて, 使用メモリ量決定アルゴリズムによって得られたリソース量で十分にノード更新処理が行えることの証明を与えた。
- Emfrp^{BCT} から C 言語への変換系を実装し, 評価実験として, 型検査や使用メモリ量の決定にかかる時間の測定, 再帰データ型を使用することによる実行時の時間的, 空間的オーバーヘッドの測定を行った。

(4) その他の成果

組込みシステムにおいてもマルチコアや GPU の利用が一般的になっており, そういった実行環境を有効利用する目的で FRP プログラムの並列実行機構および GPGPU 上での実行機構を提案し, それぞれ実装して評価を行い, その有効性を示した。

多くの組込みシステムでは実時間性が求められる。特に周期的タスクはさまざまな場面で利用されている。Emfrp においてもクロックを入力時変値とすることで周期的タスクを簡潔に記述することは可能であるが, 実行効率は必ずしも高くない。我々は時変値に更新周期の情報を付加することで周期的タスクを FRP 上で簡潔かつ高効率に実現する手法を提案した。

上記 (1) において Distributed-XFRP を用いて WSN を表現する手法について述べているが, 我々はより簡潔に WSN を表現する手法として, WSN 向けマクロ言語を提案した。マクロ言語とは無線センサーネットワーク (WSN) の開発を支援するために提案された手法であり, センサーノードシンクノード (システム全体からのデータを集約するノード) で構成される WSN 総体の動作を一つのプログラムとして記述し, コンパイラによって個々のノードで動作するコードを生成するものである。これによってノードの種類ごとにコードを開発する必要がなくなるという利点がある。我々はこの考え方を WSN に適用し, FRP にもとづくマクロ言語を設計・実装した。

5. 主な発表論文等

〔雑誌論文〕 計4件（うち査読付論文 4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 Sheng Wang, Takuo Watanabe	4. 巻 850
2. 論文標題 Functional Reactive EDSL with Asynchronous Execution for Resource-Constrained Embedded Systems	5. 発行年 2019年
3. 雑誌名 Studies in Computational Intelligence (Springer)	6. 最初と最後の頁 171-190
掲載論文のDOI (デジタルオブジェクト識別子) 10.1007/978-3-030-26428-4_12	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 松村有倫, 渡部卓雄	4. 巻 13(2)
2. 論文標題 組込みシステム向けFRP言語における状態依存動作のための抽象化機構	5. 発行年 2020年
3. 雑誌名 情報処理学会論文誌 (プログラミング)	6. 最初と最後の頁 1-13
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 後藤司, 森口草介, 渡部卓雄	4. 巻 38(2)
2. 論文標題 WSAN向けマクロプログラミング言語の提案	5. 発行年 2021年
3. 雑誌名 コンピュータ ソフトウェア	6. 最初と最後の頁 20-26
掲載論文のDOI (デジタルオブジェクト識別子) 10.11309/jssst.38.2_20	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 Akihiko Yokoyama, Sosuke Moriguchi, Takuo Watanabe	4. 巻 29
2. 論文標題 A Functional Reactive Programming Language for Small-Scale Embedded Systems with Recursive Data Types	5. 発行年 2021年
3. 雑誌名 Journal of Information Processing	6. 最初と最後の頁 685-706
掲載論文のDOI (デジタルオブジェクト識別子) 10.2197/ipsjip.29.685	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 -

〔学会発表〕 計20件（うち招待講演 0件 / うち国際学会 5件）

1. 発表者名 Takuo Watanabe
2. 発表標題 A Simple Context-Oriented Programming Extension to an FRP Language for Small-Scale Embedded Systems
3. 学会等名 10th International Workshop on Context-Oriented Programming (COP 2018), pp. 23-30, ACM [査読あり]. (国際学会)
4. 発表年 2018年

1. 発表者名 渡部卓雄
2. 発表標題 小規模組込みシステム向けFRP言語への文脈指向プログラミング機構の導入（ポスター）
3. 学会等名 組込みシステムシンポジウム(ESS 2018), pp. 17--20, 情報処理学会 [査読なし]
4. 発表年 2018年

1. 発表者名 Kazuhiro Shibanaï, Takuo Watanabe
2. 発表標題 Distributed Functional Reactive Programming on Actor-Based Runtime
3. 学会等名 8th International Workshop on Programming based on Actors, Agents, and Decentralized Control (AGERE 2018), pp. 13-22, ACM [査読あり] (国際学会)
4. 発表年 2018年

1. 発表者名 櫻井義孝, 渡部卓雄
2. 発表標題 組込みシステム向けFRP言語の静的スケジューリングを用いた並列化（ポスター）
3. 学会等名 第21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(PPL 2019), 日本ソフトウェア科学会 [査読なし]
4. 発表年 2019年

1. 発表者名 松村有倫, 渡部卓雄
2. 発表標題 組込みシステム向けFRP言語における動的動作のための抽象化 (ポスター)
3. 学会等名 第21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(PPL 2019), 日本ソフトウェア科学会 [査読なし]
4. 発表年 2019年

1. 発表者名 横山陽彦, 渡部卓雄
2. 発表標題 組込みシステム向けFRP言語に対する第一級関数の導入 (ポスター)
3. 学会等名 第21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(PPL 2019), 日本ソフトウェア科学会 [査読なし]
4. 発表年 2019年

1. 発表者名 渡部卓雄, 松村有倫, 横山陽彦
2. 発表標題 小規模組込みシステム向け関数リアクティブプログラミング言語のためのモジュール化機構
3. 学会等名 組込み技術とネットワークに関するワークショップ(ETNET2019), 情報処理学会研究報告, Vol. 2019-EMB-50, No. 42, 情報処理学会 [査読なし]
4. 発表年 2019年

1. 発表者名 松村有倫, 渡部卓雄
2. 発表標題 組込みシステム向けFRP言語における動的動作のための抽象化機構 (ポスター)
3. 学会等名 第51回組込みシステム研究発表会(SIGEMB51), 情報処理学会研究報告, Vol. 2019-EMB-51, No. 2 [査読なし]
4. 発表年 2019年

1. 発表者名 櫻井義孝, 渡部卓雄
2. 発表標題 小規模組込み向けFRP言語のための並列実行モデル (ポスター)
3. 学会等名 第51回組込みシステム研究発表会(SIGEMBS1), 情報処理学会研究報告, Vol. 2019-EMB-51, No. 2 [査読なし]
4. 発表年 2019年

1. 発表者名 渡部卓雄, 柴内一宏, 櫻井義孝, 松村有倫, 横山陽彦
2. 発表標題 関数リアクティブプログラミング言語によるサイバーフィジカルシステム開発支援にむけて (ポスター)
3. 学会等名 組込みシステムシンポジウム(ESS 2019), pp. 90-91, 情報処理学会 [査読なし]
4. 発表年 2019年

1. 発表者名 Yoshitaka Sakurai, Takuo Watanabe
2. 発表標題 Towards a Statically Scheduled Parallel Execution of an FRP Language for Embedded Systems
3. 学会等名 6th ACM SIGPLAN International Workshop on Reactive and Event-Based Languages and Systems (REBLS 2019), pp. 11--20, ACM [査読あり] (国際学会)
4. 発表年 2019年

1. 発表者名 横山陽彦, 森口草介, 渡部卓雄
2. 発表標題 小規模組込みシステム向けFRP言語に対する再帰的データ型の導入
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(PPL 2020), 日本ソフトウェア科学会 [査読あり]
4. 発表年 2020年

1. 発表者名 松村有倫, 渡部卓雄
2. 発表標題 状態遷移を表現する組み込みシステム向けFRP言語の設計 (ポスター)
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(PPL 2020), 日本ソフトウェア科学会 [査読なし]
4. 発表年 2020年

1. 発表者名 櫻井義孝, 森口草介, 渡部卓雄
2. 発表標題 FRPによるGPU上の計算の実現 (ポスター)
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(PPL 2020), 日本ソフトウェア科学会 [査読なし]
4. 発表年 2020年

1. 発表者名 辻裕太, 森口草介, 渡部卓雄
2. 発表標題 内部DSLとしての関数リアクティブプログラミング言語の実装手法 (ポスター)
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(PPL 2020), 日本ソフトウェア科学会 [査読なし]
4. 発表年 2020年

1. 発表者名 白鳥佑弥, 森口草介, 渡部卓雄
2. 発表標題 出力制約つき関数リアクティブシステムにおける入力センサの静的仕様推定 (ポスター)
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(PPL 2020), 日本ソフトウェア科学会 [査読なし]
4. 発表年 2020年

1. 発表者名 後藤司, 森口草介, 渡部卓雄
2. 発表標題 WSAN向けマクロプログラミング言語の提案 (ポスター)
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 (PPL 2020), 日本ソフトウェア科学会 [査読なし]
4. 発表年 2020年

1. 発表者名 辻裕太, 森口草介, 渡部卓雄
2. 発表標題 組み込みシステム向け関数リアクティブプログラミング言語の内部DSLとしての実装
3. 学会等名 第54回組み込みシステム研究発表会 (SIGEMB54), 情報処理学会研究報告, Vol. 2020-EMB-54, No. 3 [査読なし]
4. 発表年 2020年

1. 発表者名 Takuo Watanabe, Kazuhiro Shibana
2. 発表標題 Towards a Functional Reactive Programming Model for Developing WSANs
3. 学会等名 Asia Pacific Conference on Robot IoT System Development and Platform (APRIS 2020), pp. 1-5, IPSJ [査読あり] (国際学会)
4. 発表年 2020年

1. 発表者名 Yoshitaka Sakurai, Sosuke Moriguchi, Takuo Watanabe
2. 発表標題 Functional Reactive Programming for Embedded Systems with GPGPUs
3. 学会等名 10th International Conference on Software and Computer Applications (ICSCA 2021), pp. 75-80, ACM [査読あり] (国際学会)
4. 発表年 2021年

〔図書〕 計0件

〔産業財産権〕

〔その他〕

研究代表者は、国際会議APRIS2020における発表 Towards a Functional Reactive Programming Model for Developing WSANs (Takuo Watanabe, Kazuhiro Shibanaï)に関して、当該国際会議におけるBest Paper Award, および情報処理学会山下記念研究賞を受賞した。

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

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