

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

渡部 卓雄^{1,a)} 柴内 一宏^{1,t1} 櫻井 義孝¹ 松村 有倫¹ 横山 陽彦¹

概要: 副作用のない純粋な関数リアクティブプログラミング (FRP) 言語によるサイバーフィジカルシステム開発支援に関する我々の取り組みについて説明する。現在までに我々は、純粋な FRP 言語 Emfrp の設計と実装を通して、FRP が小規模組込みシステムの効果的な開発支援に寄与することを明らかにした。また、同言語をベースとした分散 FRP 言語を設計・実装し、センサーネットワークのような分散システムの記述に適していることを例を通して示した。本ポスター発表では、分散 FRP 言語および現在進行中の FRP に関する取り組みについて概説し、サイバーフィジカルシステムの開発における FRP の利用について考察する。

キーワード: 関数リアクティブプログラミング, 組込みシステム, サイバーフィジカルシステム

1. FRP 言語による組込みシステム開発

関数リアクティブプログラミング (Functional Reactive Programming, FRP) は、**時変値** (Time-Varying Value) と呼ばれる抽象化機構によって時間と共に連続的あるいは離散的に変化する値とその関係を宣言的に記述することで、組込みシステムに代表されるリアクティブシステムの効果的な記述を支援するプログラミングパラダイムである。我々は、マイクロコントローラに代表される小規模組込みシステムを対象とした関数リアクティブプログラミング言語 Emfrp [5]*¹を設計・実装し、いくつかの例題を通してその組込みシステム開発における有用性を明らかにしてきた。

Emfrp のプログラムは時変値の定義の集まりであり、全ての計算がリアクティブな動作として記述される。既存の言語に FRP のためのライブラリ (あるいは拡張) を導入した場合と異なり、リアクティブな動作 (時変値間の値の伝搬) 自体が計算の基本となっている。そのため言語自体は非常にシンプルであるが、リアクティブな動作をリソースの限られた環境で実行するため、以下のような設計方針をとった。具体的には静的型付けに加え、時変値を一級データではなく名前参照することや、高階関数、再帰の禁止といった言語設計上の制約を設けている。これにより、プログラムが利用する記憶領域の大きさは静的に決定できる。加えて、push 型と呼ばれる実行方式にもとづくシンプルな実行系を用いることで生成されるコードサイズを抑え

ている。また上記制約のうち、高階関数については記憶領域を静的に決定できるという性質を保ちつつ導入できる目処が立っている [9]。

以上により Emfrp のコードはマイクロコントローラのような小規模システムにおいても実用的な実行速度とメモリフットプリントを持つ一方、実行時情報にもとづく柔軟な振る舞いの表現において制約を持つ。例えば組込みシステムの実行において重要な、変化する環境への適応動作を適切にモジュール化して記述することは難しい。

このような問題に対処することを目的として、我々は Emfrp における自己反映計算 (reflection) の機構を提案し、ロボットの適応動作記述への応用等を通してその有効性を示した [8]。さらに、適応的な動作のモジュール化を促進するプログラミングパラダイムである**文脈指向プログラミング** (Context-Oriented Programming, COP) の機構を Emfrp に導入することで、適用動作の宣言的な記述とモジュール化を可能にした [7]。

2. FRP による分散システム開発

センサーネットワークやセンサーアクターネットワークのうちいくつかはデータフローシステムとしての側面を持ち、そのようなシステムはリアクティブシステムとして自然に記述できる。このとき、個々のセンサーノードやアクターノードの動作を前節で述べたように FRP 言語で記述した場合、センサー (アクター) ネットワーク全体としての動作も FRP で書くことができることが望ましい。

このような動機の下に、我々は Emfrp をベースとした分散 FRP 言語 Distributed-XFRP を設計・実装した [6]。

¹ 東京工業大学 情報理工学院 情報工学系

^{t1} 現在、株式会社ブレインパッド

^{a)} takuo@acm.org

*1 <https://github.com/psg-titech/emfrp>

Distributed-XFRP では、時変値の表現にアクターモデル [1] におけるアクターを、時変値間の値の伝搬に非同期メッセージを用いている。これにより FRP による分散システムの記述を可能にしている。プロトタイプの実行系を Erlang で実装し*2、いくつかのアプリケーションによる評価を通して提案手法の有効性を確認した。

FRP においては、関数プログラミングの利点である参照透明性が意味を持つためには、時変値の時間的な一貫性が保たれることが必要である。しかし分散システムには通信の遅延やメッセージの消失等により、全ての時変値について時間的一貫性を保つことは難しい。Distributed-XFRP では、データソースとなる時変値 (例えばセンサの測定値) それぞれについて、それに依存する時変値間での時間的一貫性 (single-source glitch freedom) が保たれるようにしている。複数のデータソースにまたがる時間的一貫性については、明示的に同期を行う仕組みを導入した。これにより、参照透明性を享受しつつ効率的な実装が可能になることを示した。

3. FRP による CPS 開発

サイバーフィジカルシステム (CPS) はセンサやアクチュエータを介して外界との相互作用を行うコンピュータを含む分散システムであり、組込みシステムの一般化とみなすことができる [2]。CPS のソフトウェア開発においては、一般的な分散システムとしての性質 (並行性、遅延および故障等) と同時に外界 (環境、機械および電子回路等) の物理的特性を考慮する必要がある。特に外界を含む系全体の動作は、連続的な変化と離散的な変化が混在するハイブリッドシステムとなり得る。このような系を適切にモデル化し、その動作を制御するプログラムを正しく書くことは一般に容易ではない。

FRP における時変値は連続的な変化を表現できるため、微分方程式として表現される連続的な動作の表現に向いている。一方、その不連続な変化は時変値で表現される式を (別のものに) 切り替えることで表現できる。これを第 1 節で述べた COP 拡張を用いて状態遷移をモジュール化することも可能であるが、現在われわれはその考え方を発展させ、switch と呼ばれる、ノード間の依存関係を実行時に動的に切り替えることに相当する機構の導入を検討している [3]。これによって状態に応じて動作を切り替えるようなプログラムを COP 拡張よりも容易かつ安全に表現することを可能にし、ハイブリッドシステムの動作を自然に記述することを可能にする。

加えて、Distributed-XFRP のようにマシンノード内とノード間にまたがる動作を統一的に記述できる言語を用いることで、CPS の動作を FRP で自然に表現できるように思

える。しかしこれに関しては、ハイブリッドシステムに特有の Zeno behavior*3 の問題や、分散システムに特有の故障について解決すべき問題が数多くある (Distributed-XFRP ではメッセージの到着順序の逆転や欠落等に関する一部の故障についてのみ対応している)。

現在は組込みシステム向けプロセッサもマルチコア化が進み、さらに GPU を含む SoC が一般的である。現在の Emfrp 処理系では時変値間の値の伝搬は逐次的に行われるが、現在それを並列化するための実行方式を開発している [4]。

謝辞 本研究は JSPS 科研費 18K11236 の助成を受けている。

参考文献

- [1] Agha, G.: *Actors: A Model of Concurrent Computation in Distributed Systems*, MIT Press (1986).
- [2] Alur, R.: *Principles of Cyber-Physical Systems*, MIT Press (2015).
- [3] 松村有倫, 渡部卓雄: 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), 情報処理学会研究報告 (EMB), Vol. 2019-EMB-51, No. 2, (オンライン), <http://id.nii.ac.jp/1001/00197686/> (2019).
- [4] 櫻井義孝, 渡部卓雄: 小規模組込み向け FRP の実行モデルの並列化, 情報処理学会研究報告 (EMB), Vol. 2019-EMB-51, No. 2, (オンライン), <http://id.nii.ac.jp/1001/00197685/> (2019).
- [5] Sawada, K. and Watanabe, T.: Emfrp: A Functional Reactive Programming Language for Small-Scale Embedded Systems, *MODULARITY Companion 2016: Companion Proceedings of the 15th International Conference on Modularity*, ACM, pp. 36–44 (online), doi:10.1145/2892664.2892670 (2016).
- [6] Shibana, K. and Watanabe, T.: Distributed Functional Reactive Programming on Actor-Based Runtime, *8th ACM SIGPLAN International Workshop on Programming Based on Actors, Agents, and Decentralized Control (AGERE 2018)*, ACM, pp. 13–22 (online), doi:10.1145/3281366.3281370 (2018).
- [7] Watanabe, T.: A Simple Context-Oriented Programming Extension to an FRP Language for Small-Scale Embedded Systems, *10th International Workshop on Context-Oriented Programming (COP 2018)*, ACM, pp. 23–30 (online), doi:10.1145/3242921.3242925 (2018).
- [8] Watanabe, T. and Sawada, K.: Towards Reflection in an FRP Language for Small-Scale Embedded Systems, *2nd Workshop on Live Adaptation of Software Systems (LASSY 2017), Companion to the 1st International Conference on the Art, Science and Engineering of Programming (Programming 2017)*, ACM, pp. 10:1–10:6 (online), doi:10.1145/3079368.3079387 (2017).
- [9] 横山陽彦: 組み込みシステム向け FRP 言語に対する第一級関数の導入, 東京工業大学工学部情報工学科 卒業論文 (2019).

*3 有限時間内に無限のイベントが発生するような振る舞いのこと。例えばボールを床に落としてバウンドさせると徐々にバウンドの間隔が短くなり、(理想的には) バウンドは無限回起こる。この動作を有限時間間隔でサンプリングして表現すると、サンプリング時間をどれだけ短くしてもいつかは表現できなくなり、意図しない動作を引き起こすことがある。

*2 <https://github.com/45deg/distributed-xfrp>