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・柴内一宏・櫻井義孝・松村有倫・横山陽彦（東京工業大学 情報理工学院）

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．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CPS開発支援に関する我々の取り組みとして，分散FRP言語Distributed-XFRPについて概説する．

組込みシステム向けFRP言語Emfrp

関数リアクティブプログラミング(FRP)は，時間と共に変化する値を抽象化した時変値(time-varying value)やイベントストリーム等を用いることで，組込みシステムなどのリアクティブな系の宣言的記述を支援するプログラミングパラダイムである．FRPを用いることで，コールバック，イベントループ，ポーリング等を排して組込みシステムの動作を見通しよく記述することが可能になる．

我々は，マイクロコントローラなどの小規模システムを対象とする純粋FRP言語Emfrpを設計・実装した[1]．右はEmfrpによるファンの制御プログラムとそれを図式化したものである．温度・湿度センサの計測値，それらから算出される不快指数とその閾値，およびファンの状態が時変値として表現されており，不快指数が閾値以上である間ファンを動作させる．Emfrpでは@last演算子で任意の時変値の直前の値を得ることができ，これによって状態依存の動作を表現できる．この例では不快指数が閾値近辺で変化することによるファンの状態の頻繁な変化の忌避（ヒステリシス制御）に用いている．

[1] Sawada, K. & T. Watanabe, "Emfrp: A Functional Reactive Programming Language for Small-Scale Embedded Systems", CROW 2016, ACM, 2016.

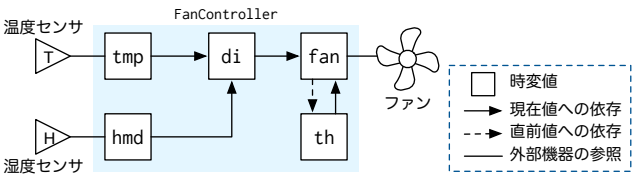
```
module FanController  # module name
in  tmp : Float,      # temperature sensor
    hmd : Float       # humidity sensor
out fan : Bool        # fan switch (GPIO)
use Std               # standard library

# discomfort (temperature-humidity) index
node di = 0.81 * tmp +
          0.01 * hmd * (0.99 * tmp - 14.3) + 46.3

# fan switch
node init[False] fan = di >= th

# threshold (w/ hysteresis behavior)
node th = 75.0 + if fan@last then -0.5 else 0.5
```

EmFrpによるファン制御プログラム

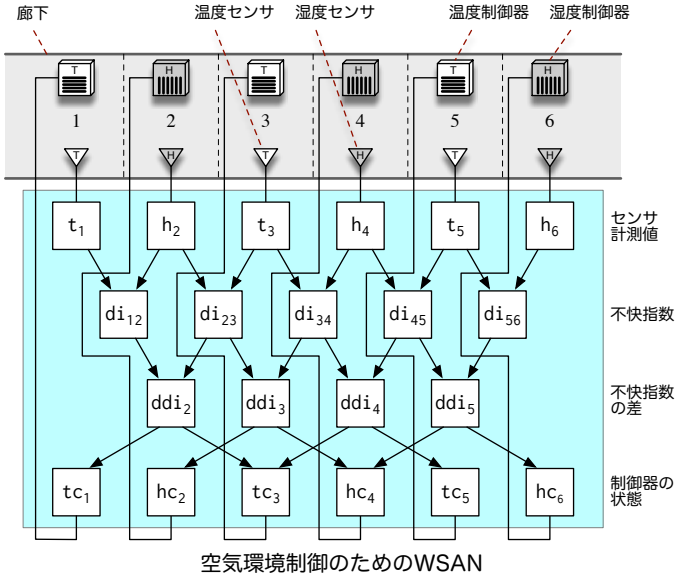


図式化したファン制御プログラム

分散FRP言語Distributed-XFRP

CPSの典型例である無線センサーアクターネットワーク(WSAN)はデータフローシステムとしての側面を持ち，リアクティブシステムとして自然に記述できる．そのため，センサー／アクターノードそれぞれの動作と同様に，WSAN総体の動作（ノード間通信等）もFRPで表現できることが望ましい．このような動機の下に，我々は分散FRP言語Distributed-XFRPを開発している[2]．この言語の実行系では，時変値としてアクターモデル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．

関数プログラミングの利点である参照透明性がFRPにおいて意味を持つためにはグリッチ（時変値の時間的な一貫性が崩れる現象）を避ける必要がある．Distributed-XFRPは，センサの計測値のような入力ノードそれぞれに対してグリッチが生じないこと(single-source glitch freedom, SSGF)を保証している．例として廊下に交互に設置されたセンサによる空気環境の制御を考える（左図）．ここで t_i (h_i)は温度（湿度）センサの計測値，di_{ij} は不快指数，ddi_i は不快指数の差，tc_i は温度（湿度）制御器の状態を表す時変値である．例えば t_3 の値は di_{23} と di_{34} に影響し，さらにこれらの値が ddi_3 に影響する．SSGFにより，t_3 の値の変化は ddi_3 に同じタイミングで伝搬する．このとき時変値が異なる計算ノード上にあってもよい．このように分散システムにおいても純粋FRPの考え方をそのまま用いることが可能になる．



空気環境制御のためのWSAN

[2] Shibanai, K. & T. Watanabe, "Distributed Functional Reactive Programming on Actor-Based Runtime", AGERE 2018, ACM, 2018.

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と同時に外界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における時変値は離散的・連続的な変化を統一的に表現できるため，ハイブリッドシステムとして表現される連続的な動作の表現に向いている．Distributed-XFRPのように計算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ことが期待できる．