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

松村 有倫　渡部 卓雄（東京工業大学）

関数リアクティブプログラミング(FRP)においてシステムの状態に応じた処理を実現するには，時変値更新のための計算を実行時に切り替える必要がある．本研究では，メモリ資源に制約のある小規模組込みシステムを対象としたFRP言語であるEmfrpにおいて，システムの状態に応じた計算の動的な切り替えを，動的にメモリを確保することなく実現するための抽象化を提案する．

関数リアクティブプログラミング(FRP)

ソフトウェアの中には，時間変化する入力に応答して計算処理を行うシステムを扱うものが存在している．典型的には，ユーザーからの入力に応答して処理を行うUIや，センサーの値に応答して制御を行うコントローラなどが挙げられる．これらのシステムのことををリアクティブシステムと呼ぶ．
関数リアクティブプログラミング(FRP)は，時間変化する値である時変値を扱うことでリアクティブシステムを記述するプログラミングパラダイムである．入力に対する応答は依存する時変値の変化に伴う値の再計算という形で実現され，時間変化する入力に対する処理を宣言的に記述することができる．

Emfrp[Sawada et al. 2016]

小規模組み込みシステムを対象としたFRP言語．
メモリ資源に制約のある環境に向けた設計となっている．

・関数が第一級オブジェクトではない
・時変値に対する計算が静的で，実行時に変化しない

これらの制約から，状態に応じた適応的動作の記述は苦手とする．

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

# discomfort index
node di = 0.81 *. tmp +. 0.01 *. hmd *. …

node fan = di >= 75.0
```

扇風機の制御を行うEmfrpプログラム

Switch拡張

状態に応じた計算の動的な切り替えのための抽象化として，Emfrpに対するSwitch拡張を提案する．これはHaskellのFRPライブラリであるYampa[Hudak et al. 2003]のswitchオペレータから発想を得たものであるが，切り替え先の計算が限定されるため，時変値の更新を行うコードを静的に用意することができる．

```
switch Example() {
  in  input1  : Int,
  out output1 : Int,
      output2 : Bool
  init StateA()

  state StateA() {
    …
    switch:
      if pred then StateB()
      else Retain
  }
  …
}
```

入力時変値と出力時変値の形式，初期状態を記述する．

状態を定義し，その状態に対する出力時変値の計算などを記述する．状態に対してパラメータを設けることもできる．

現在の状態から他の状態への遷移を記述する．

Copyright © 2019, Programming Systems Group, Department of Computer Science, Tokyo Institute of Technology

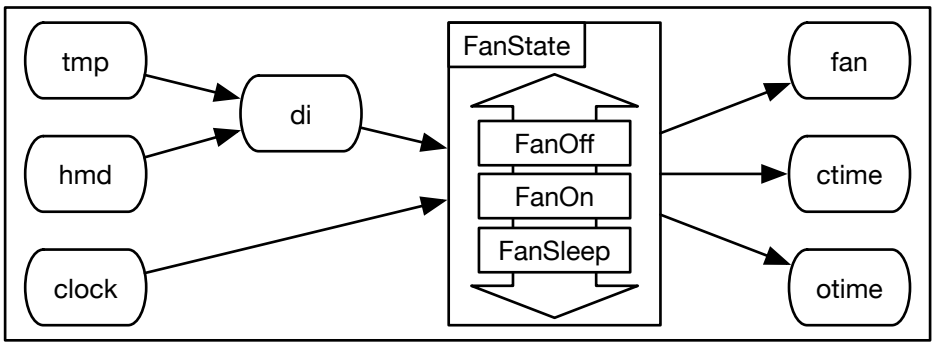
ケーススタディ

例題として，扇風機制御プログラムの拡張を考える．

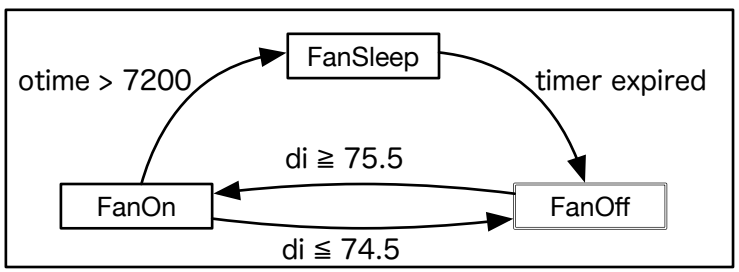
・扇風機のOn/Offだけでなく，累積稼働時間，連続稼働時間も計算する

・連続稼働時間が2時間を超えたら一定時間スリープする

扇風機に対してOff状態，On状態，Sleep状態の３つの状態を考えることができ，それぞれの状態に応じた処理を書き分けることができる．



時変値の依存関係



状態遷移（二重枠は初期状態）

```
switch FanState() {
  in  clock : Int,  # POSIX time
      di    : Float # discomfort index
  out ctime : Int,  # cumulative op. time
      otime : Int,  # continuous op. time
      fan   : Bool  # fan switch
  init FanOff()

  state FanOff() {
    …
  }
  …
}
```

入出力時変値と初期状態の定義

```
state FanOn(begin : Int) {
  node ctime = {
    dt = clock - clock@last
    ctime@last + dt
  }

  node otime = clock - begin

  node fan = True

  switch :
    if di <= 74.5 then FanOff() else
    if otime > 7200 then FanSleep(clock) else
    Retain
}
```

On状態における処理の記述

今後の展望

・リアクティブシステムにおける動的動作のユースケースの調査

・Emfrpに対する拡張の設計・実装

・実用的なアプリケーションでの実験・評価

・状態遷移（図）を取り入れた，FRPによるプログラムの設計手法の確立

関連研究

FRPにおいて適応的動作を表現するための既存の機構

switchオペレータ[Hudak et al. 2003]：

・「時変値上の関数」の時変値を扱うオペレータ

・時変値に応じて適用する関数を切り替えることで動的な動作を実現する

Emfrpに対するCOP拡張[Watanabe 2018]：

・Emfrpに文脈指向プログラミング(COP)の概念を導入したもの

・計算の切り替えを文脈に対する条件式によって指定できる

Copyright © 2019, Programming Systems Group, Department of Computer Science, Tokyo Institute of Technology