Реализация FRP библиотеки на языке Erlang

Нежевский Н.М.

Институт математики, механики и компьютерных наук им. И.И. Воровича

Кафедра информатики и вычислительного эксперимента
Научный руководитель: Брагилевский В. Н.

9 апреля 2015

- Потоки данных
- Событийность
- Преобразование данных

Обычный подход

```
a = 1; b = a + 1;
// a = 1, b = 2
a = 2;
// a = 2, b = 2
```

Реактивный подход

```
a = 1; b = a + 1;
// a = 1, b = 2
a = 2;
// a = 2, b = 3
```

Обычный подход

```
a = 1; b = a + 1;
// a = 1, b = 2
a = 2;
// a = 2, b = 2
```

Реактивный подход

```
a = 1; b = a + 1;
// a = 1, b = 2
a = 2;
// a = 2, b = 3
```

Функциональный реактивный подход

$$g(f(z(x)))$$

- Структура
- События
- Логика
- Распространение изменений

Создание и описание узлов

```
%% node() :: pid() | fun()
 {ok, Entry} = frp_api:start_node(),

 {ok, RecG}  = frp_api:start_node(fun(State, _Value) ->
                                      main:receive_new_data(State, _Value)
                                  end, {"NASDAQ", "GOOG"}),

 {ok, HandG} = frp_api:start_node(fun(State, _Value) ->
                                      main:handle_new_data(State, Value)
                                  end),

 {ok, SendDB} = frp_api:start_node(fun(State, _Value) ->
                                      receiver:send_to_db(State, Value)
                                   end),
```

Создание сети

```
%% network() :: digraph()
%% vertex()  :: unique_id() | node()

Network = frp_api:create_network(),
Nodes = [{entry, Entry}, {receiver, RecG}, {handler, HandG}, {dbsender, SendDB}],
frp_api:add_nodes(Network, Nodes),
```

Установление зависимостей

```
frp_api:add_listener(Network, entry, receiver),
frp_api:add_listener(Network, receiver, handler),
frp_api:add_listener(Network, handler, dbsender),
```

Однократное событие

```
frp_api:event({Network, entry}, get_new_data),
```

Периодическое событие

```
frp_api:timer({Network, entry}, get_new_data, 15000),
```

Краткие выводы:

- Использование модели акторов
- Callback функции определяют правила преобразования данных
- События и их обработку задаёт пользователь
- Изменение логики без изменения структуры