Programmation Synchrone

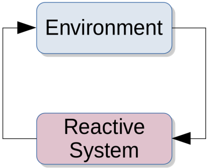
Lustre

Frédéric MALLET
Frederic.Mallet@univ-cotedazur.fr

2019-2021

Reactive systems

- Must react to all the events (stimuli) from the environment
    1. sense inputs (action, threshold, deadline)
    2. update the internal state
    3. release outputs to actuators



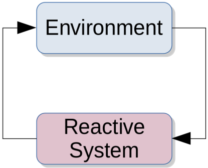
Reactive systems

- Must react to all the events (stimuli) from the environment
  1. sense inputs (action, threshold, deadline)
  2. update the internal state
  3. release outputs to actuators
- Three classes [Harel and Pnueli, 1984]

  Transformational react at the pace of inputs ;

  Interactive react at the pace of the machine (OS);

  Reactive react at the pace of the environment.



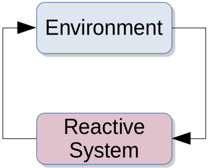
Reactive systems

- Must react to all the events (stimuli) from the environment
  1. sense inputs (action, threshold, deadline)
  2. update the internal state
  3. release outputs to actuators
- Three classes [Harel and Pnueli, 1984]

  Transformational react at the pace of inputs ;

  Interactive react at the pace of the machine (OS);

  Reactive react at the pace of the environment.
- Reactions are concurrent and timed



Synchronous paradigm

Definition (Synchrony Hypothesis)

The outputs of a system are conceptually synchronous with its inputs [Berry and Gonthier, 1992]

- Infinite loop sequence

```
for(;;) {
  read_inputs();
  process_next_state();
  write_outputs();
}
```

Synchronous paradigm

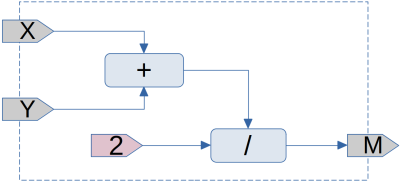
Definition (Synchrony Hypothesis)

The outputs of a system are conceptually synchronous with its inputs [Berry and Gonthier, 1992]

- Different languages and styles

Control flow	Esterel[Berry and Cosserat, 1984], SyncCharts[André, 1996], Argos[Maraninchi, 1992], Quartz
Data flow	Lustre[Caspi et al., 1987], Signal[Le Guernic et al., 1986], Scade 5
Both	Scade 6 [Colaço et al., 2017], Esterel V7 [Berry, 2007]

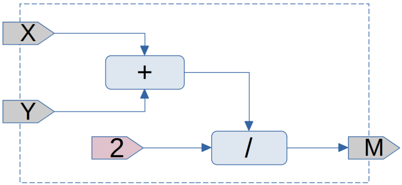
Dataflow: lustre




```
node Moyenne(X, Y : int)
  returns (M : int);
let
    M = (X + Y) / 2;
tel
```

- Classical in automatic and design of circuits (see Matlab/simulink)

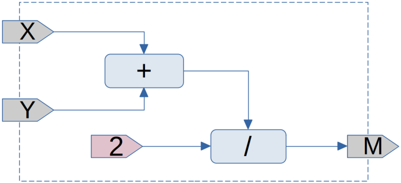
Dataflow: lustre




```
node Moyenne(X, Y : int)
  returns (M : int);
var S : int ;
let
   M = S / 2;
   S = X + Y;
tel
```

- Classical in automatic and design of circuits (see Matlab/simulink)
- S, M, X, Y, 2: (infinite) streams of integers

Dataflow: lustre



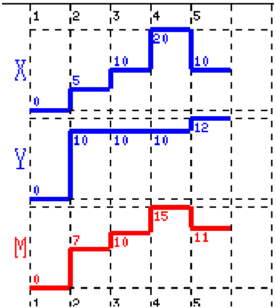

```
node Moyenne(X, Y : int)
  returns (M : int);
var S : int ;
let
   M = S / 2;
   S = X + Y;
tel
```

- Classical in automatic and design of circuits (see Matlab/simulink)
- S, M, X, Y, 2: (infinite) streams of integers
- Mathematical implementation :
  $\forall t \in \mathbb{N}, M[t] = (X[t] + Y[t])/2$

Compilation / Execution / Simulation

- Lustre V4:
  `http://www-verimag.imag.fr/The-Lustre-Toolbox.html`
- Compilation
  1. lus2ec moyenne.lus Moyenne ⟶ Moyenne.ec
  2. ec2c Moyenne.ec -v -loop ⟶ .c,.h,_loop.c
  3. gcc Moyenne.c Moyenne_loop.c -o moyenne ⟶ moyenne
- Compilation (alternative)
  - lus2c moyenne.lus Moyenne (1, 2)
  - lux moyenne.lus Moyenne (1, 2, 3) ⟶ Moyenne
- Simulation: luciole moyenne.lus Moyenne

Compilation / Execution / Simulation




```
## STEP 1 ##########
X (integer) ? 0
Y (integer) ? 5
M = 2
## STEP 2 ##########
X (integer) ? 12
Y (integer) ? 18
M = 15
## STEP 3 ##########
X (integer) ? 20
Y (integer) ? 10
M = 15
## STEP 4 ##########
X (integer) ? 12
Y (integer) ? 15
M = 13
## STEP 5 ##########
X (integer) ?
```

- Simulation: luciole moyenne.lus Moyenne

References I

André, C. (1996).
Representation and analysis of reactive behaviors: A synchronous approach.
In *CESA'96*.

Berry, G. (2007).
Circuit design and verification with Esterel v7.
In *IW. on High Level Design Validation and Test, HLDVT'07*, pages 133–136. IEEE Computer Society.

Berry, G. and Cosserat, L. (1984).
The Esterel synchronous programming language and its mathematical semantics.
Research Report RR-0327, INRIA.

Berry, G. and Gonthier, G. (1992).
The esterel synchronous programming language: Design, semantics, implementation.
Sci. Comput. Program., 19(2):87–152.

Caspi, P., Pilaud, D., Halbwachs, N., and Plaice, J. (1987).
Lustre: A declarative language for programming synchronous systems.
In *14th Annual ACM Symposium on Principles of Programming Languages*, pages 178–188. ACM Press.

Colaço, J., Pagano, B., and Pouzet, M. (2017).
SCADE 6: A formal language for embedded critical software development (invited paper).
In Mallet, F., Zhang, M., and Madelaine, E., editors, *11th International Symposium on Theoretical Aspects of Software Engineering, TASE 2017*, pages 1–11. IEEE Computer Society.

Harel, D. and Pnueli, A. (1984).
On the development of reactive systems.
In Apt, K. R., editor, *Logics and Models of Concurrent Systems*, pages 477–498. Springer.

References II

Le Guernic, P., Benveniste, A., Bournai, P., and Gautier, T. (1986).
Signal-a data flow-oriented language for signal processing.
IEEE Trans. Acoust. Speech Signal Process., 34(2):362–374.

Maraninchi, F. (1992).
Operational and compositional semantics of synchronous automaton compositions.
In Cleaveland, R., editor, *CONCUR '92, 3rd IC. on Concurrency Theory*, volume 630 of *Lecture Notes in Computer Science*, pages 550–564. Springer.