An effect type system for modular distribution of dataflow programs

Gwenaël Delaval[1] Alain Girault[1] Marc Pouzet[2]

[1]INRIA Rhône-Alpes, Pop-Art project

[2]LRI, Demons team

28/11/2006 — Synchron'06

Motivations

- Providing a language-oriented solution for the design of functionally distributed systems

Motivations

- Providing a language-oriented solution for the design of functionally distributed systems
  Alternative: separate design of each computing resource.
  Problems arised:
  - One function can involves several computing sites
    $\Longrightarrow$ separate design of closely related components, risks of unconsistencies of data.

Motivations

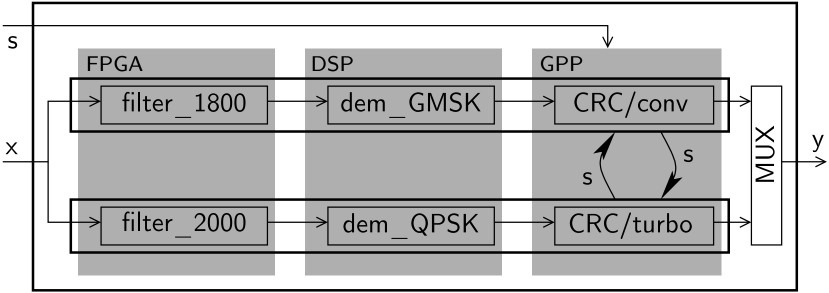
- Providing a language-oriented solution for the design of functionally distributed systems
  Alternative: separate design of each computing resource.
  Problems arised:
  - One function can involves several computing sites
    $\Longrightarrow$ separate design of closely related components, risks of unconsistencies of data.
  - One computing resource can be involved in several functions
    $\Longrightarrow$ duplicated control jeopardizing the modularity.

Motivations

- Providing a language-oriented solution for the design of functionally distributed systems
  Alternative: separate design of each computing resource.
  Problems arised:
  - One function can involves several computing sites
    $\Longrightarrow$ separate design of closely related components, risks of unconsistencies of data.
  - One computing resource can be involved in several functions
    $\Longrightarrow$ duplicated control jeopardizing the modularity.
- Performing modular distribution :
  - to avoid inlining everything
  - to allow, by mean of high-order features, dynamic reconfiguration of a site by another (by sending functions through channels)

Example

Multichannel reception system of a software-defined radio



Reception channel composed of three components:

- a pass-band filter implemented on a FPGA;
- a demodulator implemented on a DSP;
- further processing (e.g., error correction) on a general-purpose processor.

Distribution of synchronous dataflows programs

Principles

- Allowing the expression of the localization of computations: `do y = f(x) at P;`
- inference of the localization of each value and computation from expressed ones: "coloration" of the program. (P. Caspi, A. Girault and D. Pilaud, 1999)

Distribution of synchronous dataflows programs

Principles

- Allowing the expression of the localization of computations: `do y = f(x) at P;`
- inference of the localization of each value and computation from expressed ones: "coloration" of the program. (P. Caspi, A. Girault and D. Pilaud, 1999)

Realisation

- Localization of a value $\leftrightarrow$ spatial type of this value
- "coloration" $\leftrightarrow$ type inference

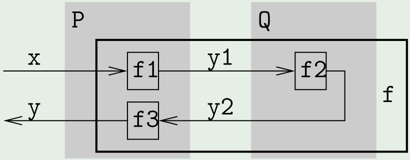
Spatial types: overview

Type and effect system (J.-P. Talpin and P. Jouvelot, 1992):

- Atomic values: the spatial type of a stream is the set of sites where this stream is present;
- functional values: spatial types of the form $s_i \prec\langle S, T\rangle\!\!\rightarrow s_o$ where
  - s_i is the spatial type of the function's inputs;
  - s_o is the spatial type of the function's outputs;
  - S is the set of sites involved in the computation of the function (used for type inference: comprises every site of s_i, s_o and T);
  - T is a set of communication channels involved in the computation (used for distribution).

Example of spatial types

```
let node f(x) = y where
  do y1 = f1(x) at P;
  do y2 = f2(y1) at Q;
  do y = f3(y2) at P
```



Assuming that `f1`, `f2` and `f3` are of spatial types $\forall\alpha.\{\alpha\} \prec\!\langle\{\alpha\},\emptyset\rangle\!\succ \{\alpha\}$, within `f`:

- `f1` and `f3` get the spatial type $\{\mathtt{P}\} \prec\!\langle\{\mathtt{P}\},\emptyset\rangle\!\succ \{\mathtt{P}\}$;
- `f2` gets the spatial type $\{\mathtt{Q}\} \prec\!\langle\{\mathtt{Q}\},\emptyset\rangle\!\succ \{\mathtt{Q}\}$
- because of the presence of the two communications, `f` is finally of spatial type $\{\mathtt{P}\} \prec\!\langle\{\mathtt{P},\mathtt{Q}\},[c_1 : \mathtt{P} \mapsto \mathtt{Q}, c_2 : \mathtt{Q} \mapsto \mathtt{P}]\rangle\!\succ \{\mathtt{P}\}$

SDR example

```
let node multichannel_sdr (x,s) = y where
    do p = gsm_or_umts(s) at GPP;
    match p with
    | Gsm ->
       do f = filter_bp_1800(x) at FPGA;
       do d = demod_GMSK(f) at DSP;
       do y = crc_conv(d) at GPP;
    | Umts ->
       do f = filter_bp_2000(x) at FPGA;
       do d = demod_QPSK(f) at DSP;
       do c = crc_conv(d) ; y = turbo_code(c) at GPP;
```

Architecture description

Architecture description: composed of site declarations as symbolic names, with encapsulation and links between sites.

```
site FPGA;
site DSP;
site GPP;
link FPGA to DSP;
link DSP to GPP;
```

Architecture description

Architecture description: composed of site declarations as symbolic names, with encapsulation and links between sites.

```
site FPGA;
site DSP;
site GPP;
link FPGA to DSP;
link DSP to GPP;
```

The architecture description defines two relations:

- a hierarchy relation $\prec_{\mathcal{R}}$: $L_1 \prec_{\mathcal{R}} L_2$ iff L_1 is a subsite of L_2
- a communication relation noted $\mapsto_{\mathcal{C}}$: $L_1 \mapsto_{\mathcal{C}} L_2$ iff there exists a communication link from L_1 to L_2.

Spatial type expressions

$$
\begin{aligned}
H &::= [x_1 : \rho_1, \ldots, x_n : \rho_n] \\
\rho &::= \forall \alpha_1, \ldots, \alpha_n . s \\
s &::= S \mid s \prec\!\langle S, T \rangle\!\succ s \mid s \times s \\
S &::= \{\ell_1, \ldots, \ell_n\} \\
\ell &::= L \mid \alpha \\
T &::= [c_1 : L_1 \mapsto L'_1, \ldots, c_n : L_n \mapsto L'_n]
\end{aligned}
$$

Judgments of the form $H \vdash e : s/S, T$: in the environment H, e is of "spatial type" s, and computing the expression e

- involves the set of sites S;
- involves the set of named communication channels T.

Subtyping

Subtyping defines an order on spatial types ($\prec$) : an expression of type s can be used wherever a more general type s' is expected.

$$\frac{H \vdash e : s/S, T \quad T' \vdash s \prec s'}{H \vdash e : s'/S, T.T'}$$

Subtyping

Subtyping defines an order on spatial types ($\prec$) : an expression of type s can be used wherever a more general type s' is expected.

$$\frac{H \vdash e : s/S, T \quad T' \vdash s \prec s'}{H \vdash e : s'/S, T.T'}$$

Example: a value of spatial type S can be used as a value of any spatial type S', subset of S.

$$\frac{S' \subset S}{\emptyset \vdash S \prec S'}$$

Subtyping and communications

Subtyping is used to handle typing of communicated values: if the set of communication links T contains a link from L_1 to L_2, then any value of spatial type $\{L_1\}$ can be used as a value of spatial type $\{L_1, L_2\}$.

$$\frac{L_1 \mapsto_{\mathcal{C}} L_2}{c : L_1 \mapsto L_2 \vdash S \cup \{L_1\} \prec S \cup \{L_1, L_2\}}$$

Typing rules: localization

A $\texttt{do}$ D $\texttt{at}$ L declaration is typed by constraining the computation of the declaration D to involve at most the site L (or any of its subsites).

$$\frac{H \vdash D : H'/S, T \quad \forall L' \in S, L' \prec_{\mathcal{R}} L}{H \vdash \texttt{do } D \texttt{ at } L : H'/S, T}$$

Typing rules: control structures

A `match` e `with` ... declaration will be duplicated on every site involved in the declarations D_i: the expression e must be present on all these sites.

$$\frac{H \vdash e : S/S', T \quad H \vdash D_i : H_i/S_i, T_i \quad S = \bigcup_{i=1}^{n} S_i}{H \vdash \begin{array}{l}\texttt{match}\ e\ \texttt{with}\\ C_1 \rightarrow D_1\\ \ldots\\ C_n \rightarrow D_n\end{array} : H_1 \uplus \ldots \uplus H_n / S \cup S', T.(T_1. \ldots . T_n)}$$

Typing rules: node definitions and applications

Applications: channels are renamed so as to allow multiple instanciations.

$$\frac{\begin{array}{c} H \vdash e_1 : s_1 \multimap\!\langle S, T\rangle\!\rightarrow s_2 / S_1, T_1 \\ H \vdash e_2 : s_1 / S_2, T_2 \\ T' = T[c'_1/c_1, \dots, c'_n/c_n] \end{array}}{H \vdash e_1(e_2) : s_2 / S_1 \cup S_2 \cup S, T_1.T_2.T'}$$

$$\frac{H, x : s \vdash e : s' / S, T}{H \vdash \texttt{let node } f(x) = e : [\mathrm{gen}_H(s \multimap\!\langle S, T\rangle\!\rightarrow s')/f]/\emptyset, \emptyset}$$

Distribution
Principle

Definition of an operation of *projection*:

$$H \vdash D : H'/S, T \xrightarrow{L} D'$$

The declaration D, projected on the site L, results in a new declaration D'

Distribution
Principle

Definition of an operation of *projection*:

$$H \vdash D : H'/S, T \stackrel{L}{\Longrightarrow} D'$$

The declaration D, projected on the site L, results in a new declaration D'

Communication channels used between two projected declarations are added as inputs or outputs streams

Projection of expressions (subtyping rules)

"Forgetting" a site results in the same expression:

$$\frac{S' \subset S}{\emptyset \vdash e : S \prec S' \xrightarrow{L} e, \emptyset}$$

Projection of expressions (subtyping rules)

"Forgetting" a site results in the same expression:

$$\frac{S' \subset S}{\emptyset \vdash e : S \prec S' \xRightarrow{L} e, \emptyset}$$

Using a channel c for communication between two sites results in the definition and use of a stream named as c.

$$\frac{L_1 \mapsto_{\mathcal{C}} L_2}{c : L_1 \mapsto L_2 \vdash e : S \cup \{L_1\} \prec S \cup \{L_1, L_2\} \xRightarrow{L_1} c, c = e}$$

$$\frac{L_1 \mapsto_{\mathcal{C}} L_2}{c : L_1 \mapsto L_2 \vdash e : S \cup \{L_1\} \prec S \cup \{L_1, L_2\} \xRightarrow{L_2} c}$$

Duplication of control structures

A `match/with` structure is duplicated on every site where needed:

$$\frac{H \vdash e : S/S', T \stackrel{L}{\Longrightarrow} e', D \quad L \in S \qquad H \vdash D_i : H_i/S_i, T_i \stackrel{L}{\Longrightarrow} D'_i \quad S = \bigcup_{i=1}^{n} S_i}{H \vdash \boxed{\begin{array}{l}\texttt{match } e \texttt{ with}\\ C_1 \rightarrow D_1\\ \ldots\\ C_n \rightarrow D_n\end{array}} : H_1 \uplus \ldots \uplus H_n / S \cup S', T.T_1.\ldots.T_n \stackrel{L}{\Longrightarrow} D; \boxed{\begin{array}{l}\texttt{match } e' \texttt{ with}\\ C_1 \rightarrow D'_1\\ \ldots\\ C_n \rightarrow D'_n\end{array}}}$$

Duplication of control structures

A `match`/`with` structure is duplicated on every site where needed:

$$\frac{\begin{array}{c} H \vdash e : S/S', T \stackrel{L}{\Longrightarrow} e', D \quad L \notin S \\ H \vdash D_i : H_i/S_i, T_i \stackrel{L}{\Longrightarrow} D'_i \quad S = \bigcup_{i=1}^{n} S_i \end{array}}{H \vdash D; \boxed{\begin{array}{l} \texttt{match } e \texttt{ with} \\ C_1 \rightarrow D_1 \\ \ldots \\ C_n \rightarrow D_n \end{array}} : H_1 \uplus \ldots \uplus H_n / S \cup S', T.T_1.\ldots.T_n \stackrel{L}{\Longrightarrow} \emptyset}$$

Modular distribution: principle

Channel names, defined or used as streams in the body of a function, will be added to the signature of this function.

Modular distribution: principle

Channel names, defined or used as streams in the body of a function, will be added to the signature of this function.

This allows multiple instanciations of functions.

Distribution example: node distribution

```
site A;
site B;
let node f(x) = z where
  do y = x + 1 at A;
  do z = y + 2 at B
```

Distribution example: node distribution

```
site A;
site B;
let node f(x) = z where
  do y = x + 1 at A;
  do z = y + 2 at B
```

spatial type of `f`: $\{A\} \prec\langle\{A,B\},[c:A\mapsto B]\rangle\!\rightarrow \{B\}$

Distribution example: node distribution

```
site A;
site B;
let node f(x) = z where
  do y = x + 1 at A;
  do z = y + 2 at B
```

spatial type of f: $\{A\} \prec\langle\{A,B\},[c : A \mapsto B]\rangle\!\!\succ \{B\}$

$\longrightarrow$ On A, c is added as an output:

```
let node f(x) = (d,c) where
  y = x + 1;
  c = y
```

Distribution example: node distribution

```
site A;
site B;
let node f(x) = z where
  do y = x + 1 at A;
  do z = y + 2 at B
```

spatial type of `f`: $\{A\} \multimap\langle\{A,B\},[c : A \mapsto B]\rangle\!\!\rightarrow \{B\}$

$\longrightarrow$ On B, c is added as an input:

```
let node f (x,c) = z where
  z = c + 2
```

Distribution example: application

```
let node g (x1,x2) = (y1,y2) where
  y1 = f(x1);
  y2 = f(x2)
```

spatial type of g:

$$\{A\} \times \{A\} \prec\langle\{A, B\}, \{c_1 : A \mapsto B, c_2 : A \mapsto B\}\rangle\!\!\rightarrow \{B\} \times \{B\}$$

Distribution example: application

```
let node g (x1,x2) = (y1,y2) where
  y1 = f(x1);
  y2 = f(x2)
```

spatial type of g:

$$\{A\} \times \{A\} \prec\!\langle\{A, B\}, \{c_1 : A \mapsto B, c_2 : A \mapsto B\}\rangle\!\rightarrowtail \{B\} \times \{B\}$$

$\Rightarrow$ use of *two distinct channels*: one for each application

Application: projection on A and B

- On A:
  - Channels c_1 and c_2 as outputs of `f`;
  - c_1 and c_2 added as outputs of `g`

```
let node g (x1,x2) = (d,d,c1,c2) where
  d,c1 = f(x1);
  d,c2 = f(x2)
```

Application: projection on A and B

- On A:
  - Channels c_1 and c_2 as outputs of `f`;
  - c_1 and c_2 added as outputs of `g`

```
let node g (x1,x2) = (d,d,c1,c2) where
  d,c1 = f(x1);
  d,c2 = f(x2)
```

- On B:
  - Channels c_1 and c_2 as inputs of `f`;
  - c_1 and c_2 added as inputs of `g`

```
let node g (x1,x2,c1,c2) = (y1,y2) where
  y1 = f(x1,c1);
  y2 = f(x2,c2)
```

Application rule

Addition of outputs and inputs channels as output and input of the node:

$$\frac{\begin{array}{c} H \vdash e_1 : s_1 \prec\langle S, T\rangle\succ s_2 / S_1, T_1 \stackrel{L}{\Longrightarrow} e_1', D_1 \\ H \vdash e_2 : s_1 / S_2, T_2 \stackrel{L}{\Longrightarrow} e_2', D_2 \\ T' = T[c_1'/c_1, \ldots, c_n'/c_n] \\ [c_1^o, \ldots, c_m^o] = [c \text{ appearing as channels whose source is L}] \\ [c_1^i, \ldots, c_p^i] = [c \text{ appearing as channels whose target is L}] \end{array}}{\begin{array}{c} H \vdash e_1(e_2) : s_2 / S_1 \cup S_2 \cup S, T_1.T_2.T' \\ \stackrel{L}{\Longrightarrow} x, D_1 ; D_2 ; (x, c_1^o, \ldots, c_m^o) = e_1'(e_2', c_1^i, \ldots, c_p^i) \end{array}}$$

Node declaration

Addition of inputs and outputs channels in the signature:

$$\frac{\begin{array}{c} H, x : s \vdash e : s'/S, T \xrightarrow{L} e', D \\ [c_1^o, \ldots, c_m^o] = [c \text{ appearing as channels whose } \textcolor{red}{\text{source}} \text{ is L}] \\ [c_1^i, \ldots, c_p^i] = [c \text{ appearing as channels whose } \textcolor{red}{\text{target}} \text{ is L}] \end{array}}{\begin{array}{c} H \vdash \boxed{\begin{array}{r} \texttt{let node } f(x) \\ = e \end{array}} : [\mathrm{gen}_H(s \prec\!\langle S, T\rangle\!\!\rightarrowtail s')/f]/\emptyset, \emptyset \\ \xrightarrow{L} \boxed{\begin{array}{r} \texttt{let node } f(x, c_1^i, \ldots, c_p^i) \\ = \texttt{let } D \texttt{ in } (e, c_1^o, \ldots, c_m^o) \end{array}} \end{array}}$$

Conclusion

- Contribution:
  - Proposal of a method for modular distribution of dataflow programs, expressed in a language extended with distribution primitives.

Conclusion

- Contribution:
  - Proposal of a method for modular distribution of dataflow programs, expressed in a language extended with distribution primitives.
- Future work:
  - How to handle polymorphism ? (i.e., functions being computed anywhere): simple when such functions involves only one site, but with several sites ?
  - Finer architecture description: typed channels,. . .
  - Examination of higher-order dataflow programs distribution. Goal: expression of dynamic reconfiguration of a site by sending functions through channels.

Thanks !...

Warning ! Appendix...

Typing rules (expressions)

$$H \vdash i : s / \mathsf{sites}(s), \emptyset$$

$$\frac{s \leq H(x)}{H \vdash x : s / \mathsf{sites}(s), \emptyset} \qquad \frac{H \vdash e_1 : s_1 / S_1, T_1 \quad H \vdash e_2 : s_2 / S_2, T_2}{H \vdash e_1, e_2 : s_1 \times s_2 / S_1 \cup S_2, T_1 . T_2}$$

$$\frac{\begin{array}{c} H \vdash e_1 : s_1 \prec\!\langle S, T \rangle\!\rightarrowtail s_2 / S_1, T_1 \\ H \vdash e_2 : s_1 / S_2, T_2 \\ T' = T[c'_1/c_1, \dots, c'_n/c_n] \end{array}}{H \vdash e_1(e_2) : s_2 / S_1 \cup S_2 \cup S, T_1 . T_2 . T'}$$

$$\frac{H \vdash D : H' / S_1, T_1 \quad H, H' \vdash e : s / S_2, T_2}{H \vdash \texttt{let}\ D\ \texttt{in}\ e : s / S_1 \cup S_2, T_1 . T_2}$$

Typing rules (declarations)

$$\frac{H \vdash e : s/S, T}{H \vdash x = e : [s/x]/S, T}$$

$$\frac{H, H_2 \vdash D_1 : H_1/S_1, T_1 \quad H, H_1 \vdash D_2 : H_2/S_2, T_2}{H \vdash D_1 \,; D_2 : H_1 \oplus H_2/S_1 \cup S_2, T_1 . T_2}$$

$$\frac{H, x : s \vdash e : s'/S, T}{H \vdash \texttt{let node } f(x) = e : [\mathrm{gen}_H(s \prec\!\langle S, T\rangle\!\succ s')/f]/\emptyset, \emptyset}$$

Projection of expressions

$$\frac{H \vdash e : s/S, T \quad T' \vdash e : s \prec s' \overset{L}{\Longrightarrow} e_2, D_2}{H \vdash e : s'/S, T.T' \overset{L}{\Longrightarrow} e_2, D_1 ; D_2}$$

Projection of expressions

$$\frac{H \vdash e : s/S, T \quad T' \vdash e : s \prec s' \xrightarrow{L} e_2, D_2}{H \vdash e : s'/S, T.T' \xrightarrow{L} e_2, D_1 ; D_2}$$

$$\frac{L \in \mathrm{sites}(s)}{H \vdash i : s/\,\mathrm{sites}(S), \emptyset \xrightarrow{L} i, \emptyset} \qquad \frac{L \notin \mathrm{sites}(s)}{H \vdash i : s/\,\mathrm{sites}(S), \emptyset \xrightarrow{L} d, \emptyset}$$

Projection of expressions

$$\frac{H \vdash e : s/S, T \quad T' \vdash e : s \prec s' \xrightarrow{L} e_2, D_2}{H \vdash e : s'/S, T.T' \xrightarrow{L} e_2, D_1 ; D_2}$$

$$\frac{L \in \mathrm{sites}(s)}{H \vdash i : s/\,\mathrm{sites}(S), \emptyset \xrightarrow{L} i, \emptyset} \qquad \frac{L \notin \mathrm{sites}(s)}{H \vdash i : s/\,\mathrm{sites}(S), \emptyset \xrightarrow{L} d, \emptyset}$$

$$\frac{L \in S}{H \vdash x : s/S, \emptyset \xrightarrow{L} x, \emptyset} \qquad \frac{L \notin S}{H \vdash x : s/S, \emptyset \xrightarrow{L} d, \emptyset}$$

Projection of expressions

$$\frac{H \vdash e : s/S, T \quad T' \vdash e : s \prec s' \stackrel{L}{\Longrightarrow} e_2, D_2}{H \vdash e : s'/S, T.T' \stackrel{L}{\Longrightarrow} e_2, D_1 ; D_2}$$

$$\frac{L \in \text{sites}(s)}{H \vdash i : s/\,\text{sites}(S), \emptyset \stackrel{L}{\Longrightarrow} i, \emptyset} \qquad \frac{L \notin \text{sites}(s)}{H \vdash i : s/\,\text{sites}(S), \emptyset \stackrel{L}{\Longrightarrow} d, \emptyset}$$

$$\frac{L \in S}{H \vdash x : s/S, \emptyset \stackrel{L}{\Longrightarrow} x, \emptyset} \qquad \frac{L \notin S}{H \vdash x : s/S, \emptyset \stackrel{L}{\Longrightarrow} d, \emptyset}$$

$$\frac{H \vdash e_1 : s_1/S_1, T_1 \stackrel{L}{\Longrightarrow} e'_1, D_1 \quad H \vdash e_2 : s_2/S_2, T_2 \stackrel{L}{\Longrightarrow} e'_2, D_2}{H \vdash e_1, e_2 : s_1 \times s_2/S_1 \cup S_2, T_1.T_2 \stackrel{L}{\Longrightarrow} (e'_1, e'_2), D_1 ; D_2}$$

Projection of declarations

$$\frac{H \vdash e : s/S, T \overset{L}{\Longrightarrow} e', D \quad L \in S}{H \vdash x = e : [s/x]/S, T \overset{L}{\Longrightarrow} D\,; x = e'} \qquad \frac{L \notin S}{H \vdash x = e : [s/x]/S, T \overset{L}{\Longrightarrow} \emptyset}$$

$$\frac{H \vdash D : H'/S, T \overset{L}{\Longrightarrow} D' \quad L \prec_{\mathcal{R}} L'}{H \vdash \texttt{do}\ D\ \texttt{at}\ L' : H'/S, T \overset{L}{\Longrightarrow} D'}$$

$$\frac{H, H_2 \vdash D_1 : H_1/S_1, T_1 \overset{L}{\Longrightarrow} D_1' \quad H, H_1 \vdash D_2 : H_2/S_2, T_2 \overset{L}{\Longrightarrow} D_2'}{H \vdash D_1\,; D_2 : H_1 \oplus H_2/S_1 \cup S_2, T_1 \oplus T_2 \overset{L}{\Longrightarrow} D_1'\,; D_2'}$$