

Intégration de données formelles dans les diagrammes d'états.

Application aux diagrammes d'états d'UML.

Pascal POIZAT^a

`poizat@lami.univ-evry.fr`

`http://www.lami.univ-evry.fr/~poizat/recherche-fr.php`

Équipe SPECIF, LaMI - UMR 8042 CNRS & Université d'Évry Val d'Essonne, GENOPOLE[®]



^aTravaux réalisés dans le cadre des collaborations Evry-Nantes (C. Attiogbé, G. Salaün) : projet SHE'S / pôle OCM

- cadre
- motivation sur un exemple
- extensions des diagrammes et sémantique
- conclusions

■ semi-formel

- + notations graphiques, lisibilité, expressivité, structuration

 - ⇒ UML (formel ?)

- outillage, cohérence ?

 - ⇒ ArgoUML, SMW, UMLAut, ...

■ formel

- + abstraction

 - ⇒ le quoi, pas le comment

- + sémantique

 - ⇒ outillage, vérification

- facilité d'apprentissage et d'emploi

- un problème : systèmes complexes
 - ⇒ structuration (objets \rightsquigarrow composants \rightsquigarrow aspects)
- des pistes :
 - ⇒ spécification orientée aspects
 - ⇒ composants (objets) de confiance
 - ⇒ comment ? \rightsquigarrow spécifications mixtes

- de nombreux langages mixtes

Cadre (abstraction – généricité)

type	dyn.	stat.	représentants
Hétérogène	AP	modèle	ObjectZ-CSP, CSP-OZ, ZCCS, ZCSP, TCOZ
	AP	alg.	LOTOS, PSF
	S/T	modèle	$\mu S\mathcal{Z}$, MaC, Event Calculus
	S/T	alg.	Korrigan , SDL, CASL-Chart, TAG
	S/T	– spéc. –	Estelle, UML , Argos, BDL
	RdP	alg.	OBJSA, Clown, CO-OPN/2
	RdP	– spéc. –	CO, OPN
Homogène	Algébrique		LTL, Rewriting Logic, ASM (famille)
	Logique		TLA, Unity, TRIO, OSL (famille)
	AP		CCS+value, CSP, π -calcul

- de nombreux langages mixtes
- apport des STS
 - ⇒ définir des “boîtes à outils formelles” génériques basées sur les STS

- une station service
- fournir différents carburants en quantité voulue
- trois pompes, trois cuves
- paiement par CB (pas encore monéo)

Une partie statique

- booléens
- entiers, réels
- carburants, pompes, cuves

Une partie statique

- booléens
- entiers, réels
- carburants, pompes, cuves

Une partie dynamique

- gestionnaire de cartes
- gestionnaire de pompes
- gestionnaire de cuves

Une partie statique

- booléens (Z)
- entiers, réels (Larch)
- carburants, pompes, cuves (Z)

Une partie dynamique (3 diag. d'états étendus)

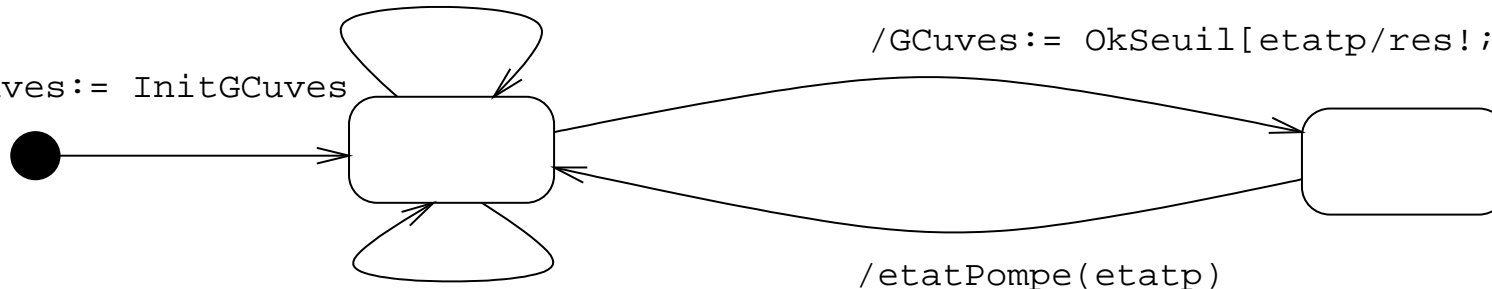
- gestionnaire de cartes
- gestionnaire de pompes
- gestionnaire de cuves

Gestionnaire Cuves

```
diminuerQt(pp: NatZ, nqt: IntZ)
```

```
/GCuves:= MajQt[pp/pp?; nqt/qtte?]
```

```
/GCuves:= InitGCuves
```



```
augmenterQt(pp: NatZ, nqt: IntZ)
```

```
/GCuves:= MajQt[pp/pp?; nqt/qtte?]
```

```
testSeuil(pa: NatZ, sp: IntZ)
```

```
/GCuves:= OkSeuil[etatp/res!; pa/pp?; sp/seuil?]
```

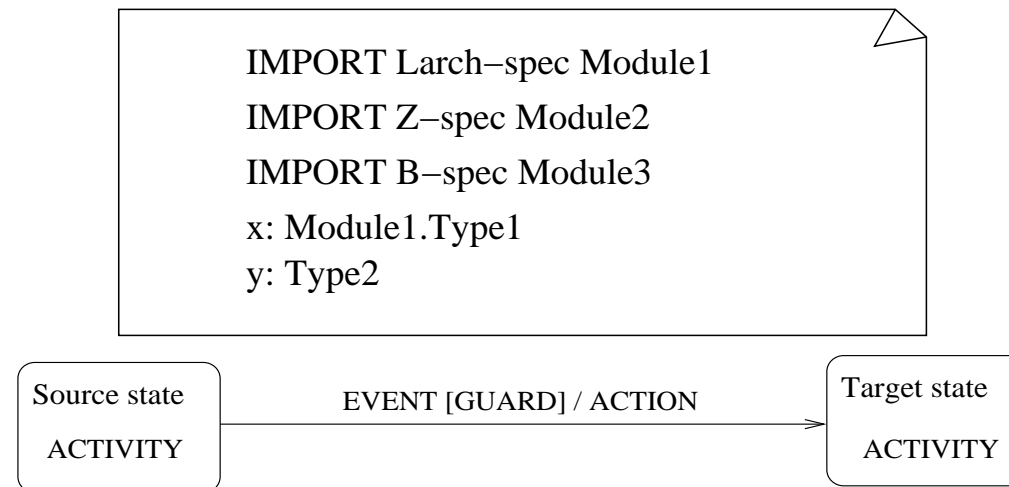
```
/etatPompe(etatp)
```

```
IMPORT Z-spec Z-donneeSE
etatp: BoolZ
```

- objectif : animer, équivalences ($=_T, \sim, \approx, \sqsubseteq_T, \sqsubseteq_F$)
- algèbres de processus
- syntaxe (opérateurs, ex : $|$)
- sémantique (structurée, ex : $\frac{P \xrightarrow{a} P' \quad Q \xrightarrow{\bar{a}} Q'}{P | Q \xrightarrow{\tau} P' | Q'}$)
- vérifications $P \sim Q \Leftrightarrow (\forall \phi \in \Phi_{HML}. P \models \phi \Leftrightarrow Q \models \phi)$
- raffinement $P \sqsubseteq_T Q$
- compositionnalité $P \sim Q \Rightarrow (\forall \mathcal{C}[\cdot]. \mathcal{C}[P] \sim \mathcal{C}[Q])$
 - ➡ travaux sur KORRIGAN
 - et plus récemment autour d'UML et diagrammes d'états



- basée sur l'expérience de plusieurs langages mixtes (Korrigan, CCS+ADT, TAG, MaC, ...)
- définir une démarche (et outils) générale d'intégration de spécifications formelles statiques (SFS) dans des spécifications formelles dynamiques (SFD)
 - ⇒ première proposition sur diagrammes d'états d'UML + SFS, communication synchrone (RR IRIN 02.03, AFADL'2003)
 - ⇒ généralisation et communication asynchrone (RR LaMI 83-2002, FASE'2003)



partie de transition	type d'interaction	exemple
EVENT	réception	$nom-evt(x_1:T_1, \dots, x_n:T_n)$
GUARD	garde	<i>prédicat</i>
ACTION	émission	$receveur \hat{ } nom-evt(t_1, \dots, t_n)$
ACTION	affectation	$x:=t$

- méta-typage
- évolution statique
- évolution dynamique et systèmes ouverts
- compositions

- nombreuses sémantiques dynamiques

- ⇒ éléments génériques

- ⇒ $event \in Q_{in}$

Contraintes :

- $\| D \|_{\text{SOS}} = \text{LTS} (\boxed{INIT}, \boxed{STATE}, \boxed{TRANS}) \Rightarrow \text{OK} !$

Notation :

- $\mathcal{D}, D = (INIT, STATE, TRANS, DeclImp, DeclVar) \in \mathcal{D}$

- $EVENT = EVENT^? \cup EVENT^!, DeclVar = DeclVar^? \cup DeclVar^!$

- $\mathcal{S} \subseteq \boxed{STATE} \times \mathcal{E} \times \boxed{Q}[EVENT^?] \times \boxed{Q}[EVENT^!]$

IMPORT X – SPEC $M \in \text{DeclImp}(D)$

$\text{def}(T, M)$

$x : T \in \text{DeclVar}(D)$

$x ::_D X$

IMPORT X – SPEC $M \in \text{DeclImp}(D)$

$\text{def}(op, M)$

$\forall i \in 1..n . \exists X . t_i ::_D X$

$op \ t_1 \ \dots \ t_n ::_D X$

$$\forall i \in 1..n . \exists X_i . t_i ::_D X_i$$

$$\exists v_i . E \vdash t_i \triangleright_{X_i} v_i$$

$$\text{act-eval}(\text{rec} \hat{e}(t_1, \dots, t_n), \langle \Gamma, E, Q_{in}, Q_{out} \rangle, D) =$$


$$\langle \Gamma, E, Q_{in}, Q_{out} \boxplus \{\text{rec} \hat{e}(v_1, \dots, v_n)\} \rangle$$

$$\exists X . t ::_D X$$


$$\exists v . E \vdash t \triangleright_X v$$

$$\text{act-eval}(x := t, \langle \Gamma, E, Q_{in}, Q_{out} \rangle, D) = \langle \Gamma, E\{x \mapsto v\}, Q_{in}, Q_{out} \rangle$$

Evaluation des termes, \triangleright_x

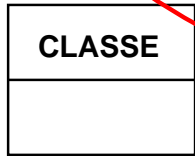
- \triangleright_{Alg} : réécriture
- $\triangleright_Z, \triangleright_B$: construction d'un LTS
- \triangleright  : ?

Evaluation des termes, \triangleright_X

- \triangleright_{Alg} : réécriture
- $\triangleright_Z, \triangleright_B$: construction d'un LTS
- \triangleright  : classes formelles, Z

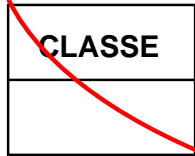
Evaluation des termes, \triangleright_x

- \triangleright_{Alg} : réécriture
- $\triangleright_Z, \triangleright_B$: construction d'un LTS
- \triangleright : classes formelles, Z



Evaluation des termes, \triangleright_x

- \triangleright_{Alg} : réécriture
- $\triangleright_Z, \triangleright_B$: construction d'un LTS
- \triangleright : classes formelles, Z



■ axiomes \rightsquigarrow règles

- ⇒ LP : $\text{dsmpos}, \text{noeq-dsmpos}$
(pas d'opérateurs a-c \Rightarrow terminaison)
- ⇒ CASL : traduction CASL \rightarrow ELAN

Construction d'un LTS ($INIT_z, STATE_z, TRANS_z$)

Construction d'un LTS ($INIT_z$, $STATE_z$, $TRANS_z$)

PointConstraint

$x : \mathbb{Z}$

$y : \mathbb{Z}$

$maxX : \mathbb{Z}$

$maxY : \mathbb{Z}$

$x \geq 0 \wedge y \geq 0 \wedge x \leq maxX \wedge y \leq maxY$

$maxX \geq 0 \wedge maxY \geq 0$

$STATE_z \subseteq \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$

$\forall (x, y, maxX, maxY) \in STATE_z, \dots$

Construction d'un LTS ($INIT_z$, $STATE_z$, $TRANS_z$)

PointConstraint

$x : \mathbb{Z}$

$y : \mathbb{Z}$

$maxX : \mathbb{Z}$

$maxY : \mathbb{Z}$

$x \geq 0 \wedge y \geq 0 \wedge x \leq maxX \wedge y \leq maxY$
 $maxX \geq 0 \wedge maxY \geq 0$

$STATE_z \subseteq \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$

$\forall (x, y, maxX, maxY) \in STATE_z, \dots$

Construction d'un LTS ($INIT_z, STATE_z, TRANS_z$)

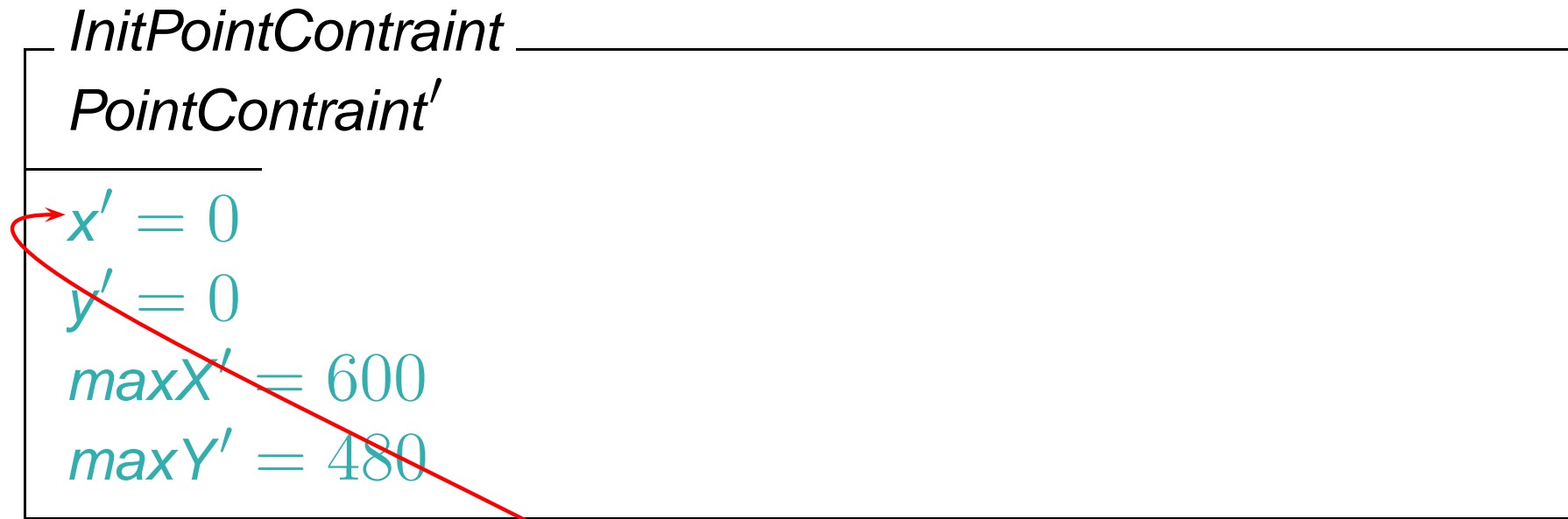
<i>InitPointConstraint</i>
<i>PointConstraint'</i>
$x' = 0$
$y' = 0$
$maxX' = 600$
$maxY' = 480$

$$INIT_z \subseteq STATE_z$$

$$\forall (x, y, maxX, maxY) \in INIT_z, \dots$$

$$INIT_z = \{(0, 0, 600, 480)\}$$

Construction d'un LTS ($INIT_z, STATE_z, TRANS_z$)



$$INIT_z \subseteq STATE_z$$

$$\forall (x, y, maxX, maxY) \in INIT_z, \dots$$

$$INIT_z = \{(0, 0, 600, 480)\}$$

Construction d'un LTS ($INIT_z, STATE_z, TRANS_z$)

Move

$\Delta PointConstraint$

$dx? : \mathbb{Z}$

$dy? : \mathbb{Z}$

$$x + dx? \geq 0 \wedge y + dy? \geq 0$$

$$x + dx? \leq maxX \wedge y + dy? \leq maxY$$

$$x' = x + dx$$

$$y' = y + dy$$

$$maxX' = maxX$$

$$maxY' = maxY$$

$$(0, 0, 600, 480) \xrightarrow{Move[10/dx?,15/dy?]} (10, 15, 600, 480)$$

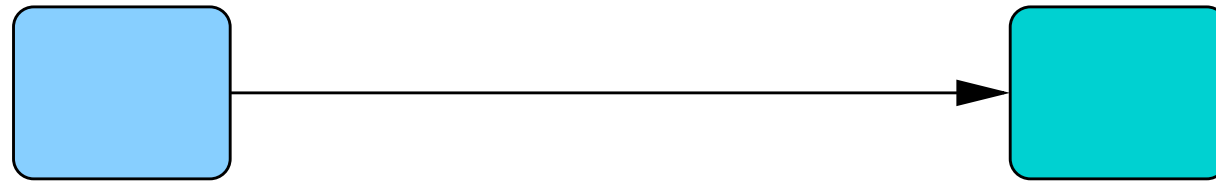
Construction d'un LTS ($INIT_z, STATE_z, TRANS_z$)

$$E \vdash I \triangleright_z s' \Leftrightarrow \exists s \subseteq E . s \xrightarrow{I} s' \in TRANS_z.$$

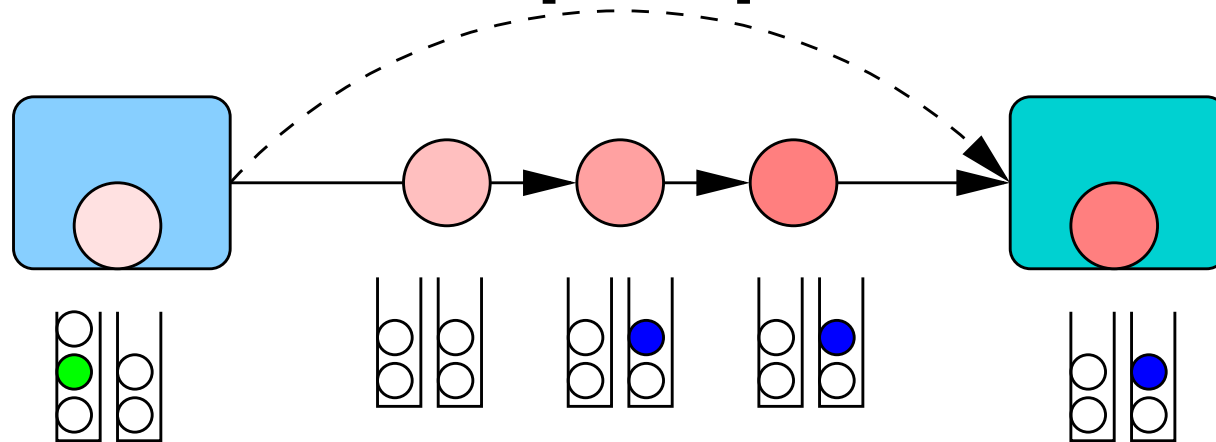
$$EVENT^{?+} = EVENT^? \cup \{\varepsilon\}$$

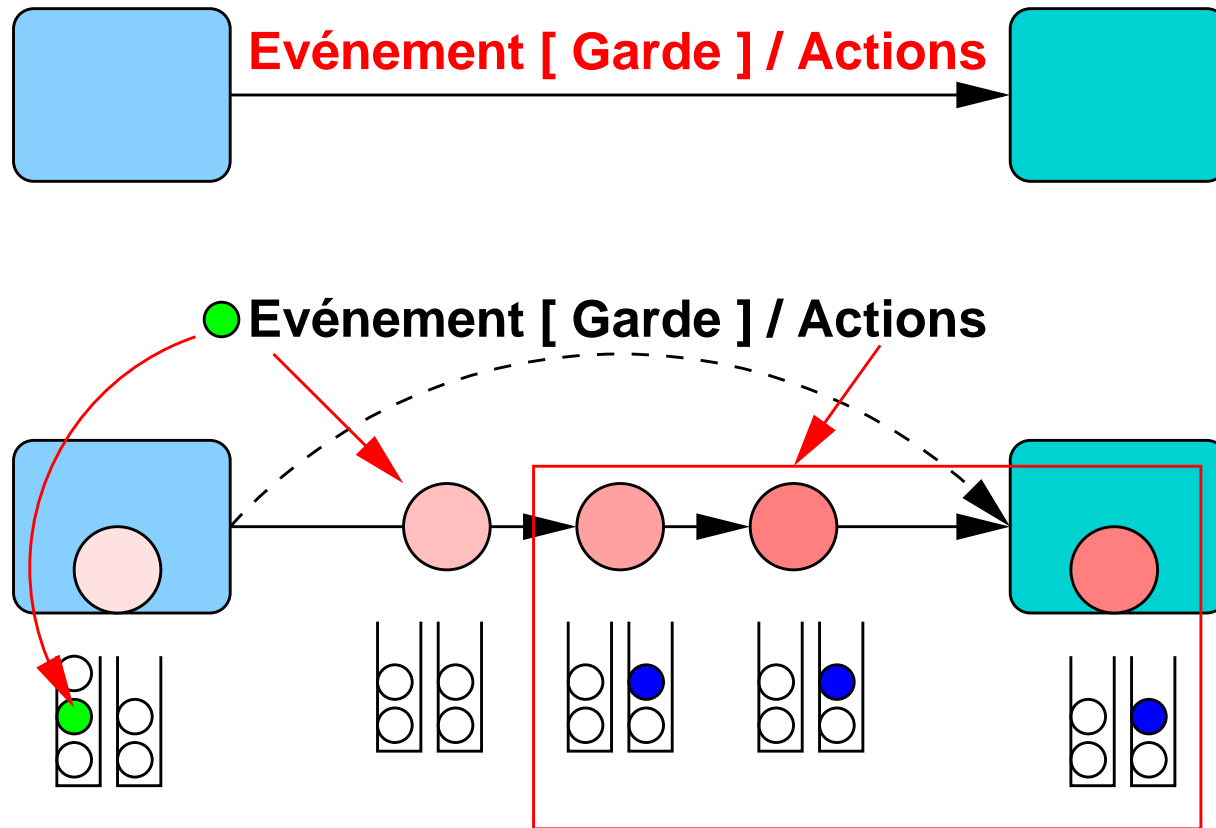
$\| D \|_{\text{SOS}} = LTS(\underline{INIT}, \underline{STATE}, \underline{TRANS})$ avec :

- $\underline{STATE} \subseteq \mathcal{S}$
- $\underline{INIT} \subseteq \underline{STATE}$
- $\underline{TRANS} \subseteq \underline{STATE} \times EVENT^{?+} \times \underline{STATE}$



● Evénement [Garde] / Actions





|| D ||_{SOS}^{open} = $LTS(\underline{INIT}^{open}, \underline{STATE}^{open}, \underline{TRANS}^{open})$ avec :

- $\underline{INIT}^{open} \subseteq \underline{INIT}$
- $\underline{TRANS}^{open} \subseteq \underline{TRANS} \times \boxed{Q}[EVENT?] \times \boxed{Q}[EVENT!]$
- $\underline{STATE}^{open} \subseteq \text{SOURCE}(\underline{TRANS}^{open}) \cup \text{TARGET}(\underline{TRANS}^{open})$

$$\langle \Gamma, E, Q_{in}, Q_{out} \rangle \xrightarrow{!} \langle \Gamma', E', Q'_{in}, Q'_{out} \rangle \in \underline{TRANS}$$

$$\exists E_{out} \subseteq Q_{out}$$

$$\exists E_{in} \subseteq \boxed{\mathcal{P}}(EVENT?)$$

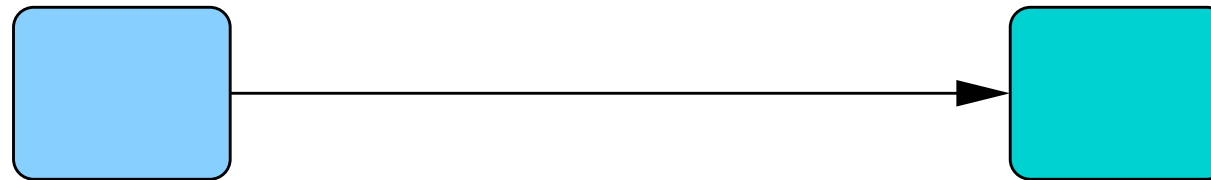
$$\langle \Gamma, E, Q_{in}, Q_{out} \rangle \xrightarrow{!}_{E_{in}, E_{out}} \langle \Gamma', E', Q'_{in} \boxed{\cup} E_{in}, Q'_{out} \boxed{\setminus} E_{out} \rangle \in \underline{TRANS}^{open}$$

|| D ||_{SOS}^{open} = $LTS(\underline{INIT}^{open}, \underline{STATE}^{open}, \underline{TRANS}^{open})$ avec :

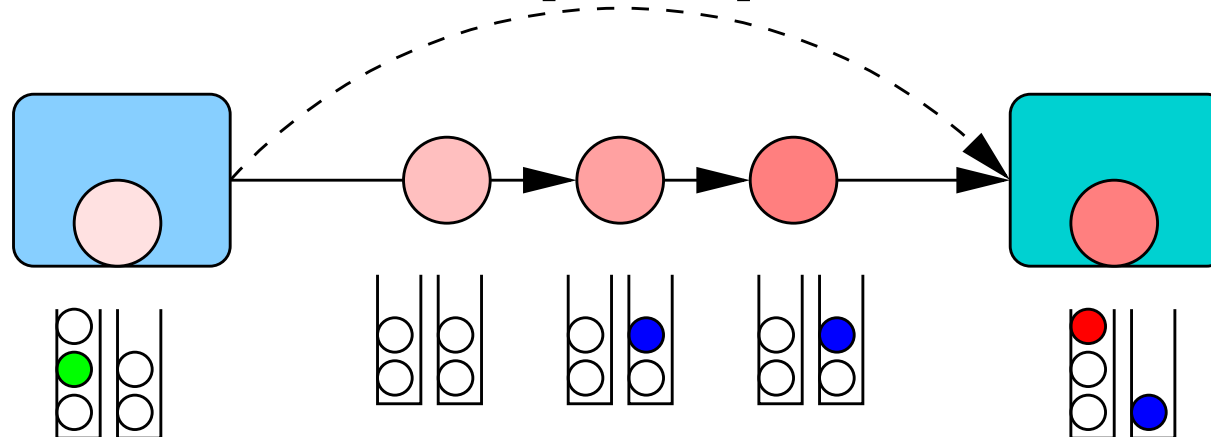
■ $\underline{INIT}^{open} \subseteq \underline{INIT}$

■ $\underline{TRANS}^{open} \subseteq \underline{TRANS} \times \boxed{Q}[EVENT?] \times \boxed{Q}[EVENT!]$

■ $\underline{STATE}^{open} \subseteq \text{SOURCE}(\underline{TRANS}^{open}) \cup \text{TARGET}(\underline{TRANS}^{open})$



● Événement [Garde] / Actions



$$\| \bigcup_{i \in 1..n} D_i \|_{\text{oper}}^{\text{open}} =$$

$LTS(\overline{INIT}^{\text{open}}(\bigcup_{i \in 1..n} D_i), \overline{STATE}^{\text{open}}(\bigcup_{i \in 1..n} D_i), \overline{TRANS}^{\text{open}}(\bigcup_{i \in 1..n} D_i))$ avec :

- $\overline{INIT}(\bigcup_{i \in 1..n} D_i) \subseteq \prod_i \overline{INIT}^{\text{open}}(D_i)$
- $\overline{TRANS}(\bigcup_{i \in 1..n} D_i) \subseteq \{t \in \prod_i \overline{TRANS}^{\text{open}}(D_i) \mid CC(t)\}$
- $\overline{STATE}(\bigcup_{i \in 1..n} D_i) \subseteq \overline{INIT}(\bigcup_{i \in 1..n} D_i) \cup \text{TARGET}(\overline{TRANS}(\bigcup_{i \in 1..n} D_i))$

$$CC(S_1 \xrightarrow{I_1}_{E_{in_1}, E_{out_1}} S'_1, \dots, S_n \xrightarrow{I_n}_{E_{in_n}, E_{out_n}} S'_n) \Leftrightarrow$$

$$\forall k \in 1 \dots n . \forall D_j \hat{e} \boxed{\in} E_{out_k} .$$

$$D_j \in \bigcup_{i \in 1..n} D_i \implies e \boxed{\in} E_{in_j}$$

Bilan

- démarche d'intégration SFS + SFD
- partiellement outillée
 - ⇒ plongement dans PVS (RR IRIN 03.02)
 - ⇒ animation et vérification de modèle (CLAP2)

Bilan

- démarche d'intégration SFS + SFD
- partiellement outillée
 - ⇒ plongement dans PVS (RR IRIN 03.02)
 - ⇒ animation et vérification de modèle (CLAP2)

Perspectives

- analyse $\parallel . \parallel$ vs. $\parallel . \parallel_{\text{SOS}}$
- intégration dans un outil UML (SMW)
- génération de code
- cohérence, suite : ST(S) + MSC

Intégration de données formelles dans les diagrammes d'états.

Application aux diagrammes d'états d'UML.

Pascal POIZAT^a

`poizat@lami.univ-evry.fr`

`http://www.lami.univ-evry.fr/~poizat/recherche-fr.php`

Équipe SPECIF, LaMI - UMR 8042 CNRS & Université d'Évry Val d'Essonne, GENOPOLE[®]



^aTravaux réalisés dans le cadre des collaborations Evry-Nantes (C. Attiogbé, G. Salaün) : projet SHE'S / pôle OCM