

# ***Les Ambients : un outil théorique pour l'ubiquité ?***

Arnaud Bailly

Laboratoire d'Informatique Fondamentale de Lille



▶ Introduction/historique

- ▶ Introduction/historique
- ▶ *Les calculs d'Ambients*

- ▶ Introduction/historique
- ▶ *Les calculs d'Ambients*
- ▶ Applications

- ▶ Introduction/historique
- ▶ *Les calculs d'Ambients*
- ▶ Applications
- ▶ Conclusion

- ▶ Langage formel pour la modélisation de processus mobiles communicants  
[Cardelli and Gordon, 2000]

## Algèbres de processus

- ▶ CSP [Brookes et al., 1984] : ensemble de traces, notion d'équivalence basée sur les *refus*
- ▶ CCS [Milner, 1992] : restriction de noms, notion d'équivalence basée sur la *bisimulation*
- ▶  $\pi$ -calcul [Milner et al., 1990] : passage de noms identifiants des *canaux de communication*
- ▶  $\pi$ -calcul d'ordre supérieur,  $\pi$ -calcul asynchrone, Blue calculus, Join calculus, ...

- ▶ Approche similaire au  $\lambda$ -calcul dans l'univers de fonctions
- ▶ Définir un noyau le plus petit possible pour pouvoir *raisonner* sur des processus mobiles (voir aussi [Bergner et al., 1999] pour un aperçu sur la question des modèles pour la mobilité)
- ▶ Intégration des notions de *concurrency* et de *localisation*

- ▶ Introduction/historique
- ▶ **Les calculs d'Ambients**
- ▶ Applications
- ▶ Conclusion

$$T ::= M \mid P$$
$$M ::= n \mid x \mid \text{in}M \mid \text{out}M \mid M.M$$
$$P ::= 0 \mid M.P \mid P|Q \mid (\nu n)P \mid !P \mid M[P] \mid (x)P \mid \langle M \rangle$$

T = terme, M = messages (capacités), P = processus.

- ▶ Congruence structurelle : équivalence de processus basée sur leur *structure statique*.
- ▶ Règles de réduction : définie comment les termes sont modifiés (réduits).
- ▶ *Bisimulations* : définition d'une relation d'équivalence sémantique entre processus en fonction des réductions possibles.

Relation d'équivalence entre termes en fonction de leur structure (comprend l'associativité et la commutativité de  $|$ ,  $0$  élément neutre):

$$!P \equiv P|!P$$

$$(M.N).P \equiv M.(N.P)$$

$$(\nu m)0 \equiv 0$$

$$(\nu n)(\nu m)P \equiv (\nu m)(\nu n)P \quad \text{si } m \neq n$$

$$P|(\nu m)Q \equiv (\nu m)(P|Q) \quad \text{si } m \text{ non libre dans } P$$

$$n[(\nu m)P] \equiv (\nu m)n[P] \quad \text{si } m \neq n$$

(entrée)	$n[\text{in}m.P Q] \mid m[R] \longrightarrow m[n[P \mid Q] \mid R]$
(sortie)	$m[n[\text{out}m.P \mid Q] \mid R] \longrightarrow n[P \mid Q] \mid m[R]$
(communication)	$(x)P \mid \langle M \rangle \longrightarrow P\{M/x\}$

Défini quelles sont les contextes dans lesquelles les règles de réduction peuvent s'appliquer.

$$P \longrightarrow Q \implies n[P] \longrightarrow n[Q]$$

$$P \longrightarrow Q \implies (\nu n)P \longrightarrow (\nu n)Q$$

$$P \longrightarrow Q \implies P \mid R \longrightarrow Q \mid R$$

- ▶ La syntaxe et la sémantique sont trop libérales et permettent des constructions absurdes :

$$\langle \text{in}m \rangle \mid (x)x[P] \longrightarrow (\text{in}m)[P]$$

- ▶ Typage permet d'attribuer un *sujet de conversation* (topic) pour chaque ambient en fonction des messages échangeables.
- ▶ Typage des noms d'ambients et des processus.
- ▶ Théorème de réduction du sujet permet de s'assurer de la correction du typage : un terme bien typé se réduit en un autre terme bien typé.

**Ambients** : définition initiale par Cardelli & Gordon.

Primitive d'ouverture d'un ambient :

$$\text{open } m.P \mid m[Q] \rightarrow P \mid Q$$

**Ambients** : définition initiale par Cardelli & Gordon.

**Safe A.** : ajoute co-actions pour chacune des primitives de mobilité.

$$\begin{array}{l}
 n[\text{in}m.P_1 \mid P_2] \mid m[\overline{\text{in}}m.Q_1 \mid Q_2] \longrightarrow m[n[P_1 \mid P_2] \mid Q_1 \mid Q_2] \\
 m[n[\text{out}m.P_1 \mid P_2] \mid \overline{\text{out}}m.Q_1 \mid Q_2] \longrightarrow n[P_1 \mid P_2] \mid m[Q_1 \mid Q_2] \\
 \text{open}n.P \mid n[\overline{\text{open}}n.Q_1 \mid Q_2] \longrightarrow P \mid Q_1 \mid Q_2
 \end{array}$$

**Ambients** : définition initiale par Cardelli & Gordon.

**Safe A.** : ajoute co-actions pour chacune des primitives de mobilité.

**Secure Safe A.** : syntaxe et sémantique idem *Safe A.*.  
différencié par les règles de typage : notion de groupes de sécurité et de règles d'accès dans une hiérarchie d'ambients.

**Ambients** : définition initiale par Cardelli & Gordon.

**Safe A.** : ajoute co-actions pour chacune des primitives de mobilité.

**Secure Safe A.** : syntaxe et sémantique idem *Safe A.*

**Boxed A.** : supprime ouverture (`open`), ajoute primitives de communication spécialisées (synchrones).

$$(x)^n P \mid n[\langle M \rangle Q \mid R] \longrightarrow P\{M/x\} \mid n[Q \mid R]$$

$$\langle M \rangle P \mid n[(x)^\uparrow Q \mid R] \longrightarrow P \mid n[Q\{M/x\} \mid R]$$

$$\langle M \rangle^n P \mid n[(x)Q \mid R] \longrightarrow P \mid n[Q\{M/x\} \mid R]$$

$$(x)P \mid n[\langle M \rangle^\uparrow Q \mid R] \longrightarrow P\{M/x\} \mid n[Q \mid R]$$

- ▶ *Logique modale* : définit modes et quantificateurs permettant d'exprimer des *propriétés* pour certains noms, certaines structures, certains processus.
- ▶ Chaque formule est *interprétée* dans les termes du calcul des ambients. Il devient alors possible de vérifier des propriétés d'ambients.
- ▶ **Problèmes** : complexité, indécidabilité de la plupart des fragments "intéressants", outils.

- ▶ Algèbre de processus récente [Schmitt and Stefani, 2004]
- ▶ Inspiré des travaux sur Fractal, M-calcul, ChAM, ...
- ▶ *Caractéristiques* :
  - ▷ Ordre supérieur
  - ▷ Paramétré par les modalités de communications (langage de pattern-matching plus ou moins expressif et puissant)

**Sécurité des systèmes/Cryptographie** Vérification de protocoles, propriétés de containment, vérification de l'environnement d'un agent.

**Sécurité des systèmes/Cryptographie** Vérification de protocoles, propriétés de containment, vérification de l'environnement d'un agent.

**Sécurité des systèmes/Cryptographie** Vérification de protocoles, propriétés de containment, vérification de l'environnement d'un agent.

**Bio-informatique** Modélisation des cellules (membranes + plasmés) et des interactions entre *compartiments biologiques (BioAmbients)*

**Sécurité des systèmes/Cryptographie** Vérification de protocoles, propriétés de containment, vérification de l'environnement d'un agent.

**Bio-informatique** Modélisation des cellules (membranes + plasmés) et des interactions entre *compartiments biologiques (BioAmbients)*

**Données semi-structurées** Utilisation logique des ambients pour raisonner sur des structures arborescentes (cf. MOSTRARE)

Deux exemples simples ([Cardelli and Gordon, 2000]):

**Firewall** Contrôle d'accès d'un agent par un firewall.

$$Agent = k'[\text{open}k.k''[Q]]$$

$$Firewall = (\nu w)w[k[\text{out}w.\text{in}k'.\text{in}w]|\text{open}k'.\text{open}k''.P]$$

Deux exemples simples ([Cardelli and Gordon, 2000]):

**Firewall** Contrôle d'accès d'un agent par un firewall.

**Routage** Routage de paquets

$$\begin{aligned} \text{paquet}(pkt) &= pkt[!(x).x|!openroute] \\ \text{route}(pkt, P, to) &= \text{route}[inpkt.\langle to \rangle \mid P \end{aligned}$$

- ▶ Introduction/historique
- ▶ *Les calculs d'Ambients*
- ▶ **Démonstration**
- ▶ Conclusion

**Ambicobjs** : modélisation dynamique et graphique dans les calculs des ambients (équipe MIMOSA, INRIA Sophia-Antipolis).

- ▶ Introduction/historique
- ▶ *Les calculs d'Ambients*
- ▶ Applications
- ▶ **Conclusion**

- ▶ Cadre formel permettant de raisonner sur les problèmes d'interactions entre un *contexte* et une *application*.
- ▶ Travaux théoriques intensifs (cf. références bibliographiques) : typage, logiques, fragments possédant certaines propriétés
- ▶ Quelques outils académiques pour raisonner sur ces langages.

**Passage à l'échelle** : outiller et offrir des modèles plus riches pour travailler sur des applications "réelles"

**Complexité** : complexité des algorithmes (bisimulation, model-checking, preuve) même sur des fragments simples

**Investissement**



## References

- [Bergner et al., 1999] Bergner, K., Grosu, R., Rausch, A., Schmidt, A., Scholz, P., and Broy, M. (1999). Focusing on mobility. In Sprague, R. H. and Jr., editors, *Proceedings of the Thirty-Second Annual Hawaii International Conference on System Sciences*. IEEE Computer Society.
- [Brookes et al., 1984] Brookes, S., Hoare, C., and Roscoe, A. (1984). A theory of communicating sequential processes. *Journal of the ACM*, 31(3):560–599.

## References

[Cardelli and Gordon, 2000] Cardelli, L. and Gordon, A. (2000). Mobile ambients. *Theoretical Computer Science*, 240(1):177–213.

[Milner, 1992] Milner, R. (1992). *Handbook of Theoretical Computer Science*, volume B - Formal Models and Semantics, chapter Operational and Algebraic Semantics of Concurrent Processes, pages 1203–1242. Elsevier, Amsterdam, Netherlands.

## References

[Milner et al., 1990] Milner, R., Parrow, J., and Walker, D. (1990). A calculus of mobile processes, part i.

[Schmitt and Stefani, 2004] Schmitt, A. and Stefani, J.-B. (2004). The Kell Calculus: A Family of Higher-Order Distributed Process Calculi. In *Proceedings of the Global Computing 2004 workshop*, Lecture Notes in Computer Science, Venice, Italy.

- ▶ *Ambient calculi online* : une bibliographie complète et à jour

<http://xdguan.freezope.org/wiki/AmbientCalculiOnline>

- ▶ *Cardelli's Ambit* : page de Luca Cardelli, avec une applet implantant le calcul des ambients en java

<http://www.luca.demon.co.uk/Ambit/Ambit.html>

- ▶ *mobilité* : page générale sur la mobilité.

<http://move.to/mobility>