Les espaces d'interaction

Vers une géométrie des systèmes d'agents mobiles

Frédéric Peschanski* — Reynald Affeldt* — Jean-Pierre Briot**

* *Université de Tokyo*

{pesch,affeldt}@yl.is.s.u-tokyo.ac.jp

** *Laboratoire d'Informatique de Paris 6*

Jean-Pierre.Briot@lip6.fr

RÉSUMÉ. *Les systèmes d'agents mobiles sont de grande complexité, notamment du fait de leur nature hautement dynamique. Au delà des expérimentations « pratiques », de nombreux travaux s'intéressent à la caractérisation plus fondamentale de ces systèmes. Mais la plupart des approches négligent certains aspects selon nous fondamentaux comme la sémantique de communication. Elles sont de plus généralement peu flexibles à ce niveau et développent en outre une vision subjective de la mobilité. Pour aborder ces questions, nous proposons le modèle des Espaces d'Interaction ($\mathcal{EI}$), une sémantique qui se distingue par son caractère géométrique. Un $\mathcal{EI}$ possède trois dimensions : agents ou localisations, canaux de communication et état de ces canaux. Les opérations fondamentales des agents, pour communiquer ou se mouvoir, sont assimilées à des transformations géométriques simples. Au dessus des $\mathcal{EI}$, nous proposons le $cube$-calcul. Conçu comme une variante appliquée du π-calcul, ce prototype de langage de programmation implémente des formes objectives de mobilité des agents et des liens de communication. Nous insistons également sur son caractère « réaliste ».*

ABSTRACT. *Mobile agent systems, highly dynamic by nature, are difficult to reason about or implement efficiently and safely. Beyond "practical" experiments, several work propose to characterize such systems in a more fundamental way. However, most approaches are somewhat inflexible and abstract away from crucial aspects of communication semantics. They also invariably develop a subjective view of mobility. In an effort to address this issues, we propose the model of Interaction Spaces ($\mathcal{I_S}$), a geometrical characterization of mobile agent systems. $\mathcal{I_S}$ provide dimensions for agents or locations, for communication channels and also for channels state. The basic operations of agents and channels, such as communicating values or moving from sites to sites, are defined as simple geometrical transformations. Above interaction spaces, we propose a prototype programming language, the $cube$-calculus. This language, designed as a derivative of the π-calculus, implements an objective form of both mobility of agents and communication links. We also insist on its "realistic" nature.*

MOTS-CLÉS : *agents mobiles, modèle géométrique, sémantique opérationnelle.*

KEYWORDS: *mobile agents, geometrical model, operational semantics*

1. Introduction

Les systèmes d'agents logiciels dénotent des interactions de grande complexité, à la fois *concurrentes*, *distribuées* et *mobiles*. De nombreuses plates-formes, prototypes de recherches ou infrastructures industrielles, permettent la description et le contrôle de ces systèmes. Au delà de ces investigations « pratiques », la compréhension plus « fondamentale » des systèmes d'agents mobiles fait l'objet de nombreux débats. Les recherches fondées sur les algèbres de processus sont en particulier fort nombreuses, et notamment celles dérivées du π-calcul [MIL 99]. La théorie de mobilité du π-calcul repose sur le concept de *mobilité virtuelle des liens* de communication entre agents. La *mobilité des agents* eux-mêmes, plus « physique », est discutée dans des versions appliquées –et localisée– du calcul (e.g. [FOU 96] [UNY 01]).

Nous voyons dans la profusion de variantes du π-calcul le signe d'un certain manque de flexibilité du modèle. L'application concrète des principes très expressifs du langage semble également de nature complexe. Le langage que nous présentons dans cet article, le $cube$-calcul, « implémente » et enrichi le π-calcul par la communication distribuée et la mobilité objective des liens et des agents. Au delà de ses propriétés syntaxiques, c'est dans la sémantique opérationnelle de ce langage que se distingue notre approche. Cette sémantique repose sur le modèle des *Espaces d'Interaction* qui adoptent un point de vue *géométrique*. La topologie des systèmes d'agent mobiles y est explicitée selon trois dimensions : (1) les localisations des agents, (2) les canaux de communication, et (3) l'état de ces canaux. Les opérations de base des agents, concernant la communication et la mobilité sont caractérisées par des transformations simples de cette géométrie. Concernant les aspects mobiles, nous proposons tout d'abord un raffinement presque directement implémentable (cf. [MAS 04]) de la mobilité des liens, en complément de la mobilité des agents, plus « standard ». Mais ce qui distingue encore plus notre approche concerne le traitement naturel de la *mobilité objective*, c'est-à-dire initiée à l'extérieur du système, en complément de sa variante subjective généralement adoptée. L'explicitation des canaux de communication, ainsi que de leur état, permet de caractériser précisément et de façon très flexible la sémantique de communication dans les espaces d'interaction ; elles est souvent figée dans les algèbres de processus. Par défaut, la sémantique proposée est à la fois asynchrone, multipoint et respectant l'ordre causal par canal de communication. Ce choix est motivé par un soucis double de « réalisme » (la sémantique choisie est implémentable efficacement) et d'expressivité (vaste domaine d'application).

Nous introduisons en section 2 la syntaxe et la sémantique informelle du $cube$-calcul pour décrire le comportement des agents communiquants et mobiles. Nous en illustrons informellement la sémantique de communication ainsi que les capacités de mobilité objective. Nous insistons particulièrement sur le caractère « réaliste » de ce langage. Nous présentons ensuite, en section 3, les espaces d'interaction, fondations géométriques sur lesquelles reposent la sémantique du $cube$-calcul. Cette sémantique se divise en une partie purement géométrique et un système de réduction servant de lien avec les algèbres de processus. Un panorama des travaux de recherche connexes est proposé en section 4. Conclusion et bibliographie terminent notre discussion.

Non-terminal	Définition	Rôle
<prog> : :=	<prefix>.<prog>	préfixes *géométriques*
	0	inerte
	<prog>$_l$	expression localisée en l
	$A(y_1, \ldots, y_n)$.<expr>	Comportement
	<prog> $+$ <prog>	Choix
	<prog> $\parallel$ <prog>	Composition parallèle
<behavior> : :=	$A(x_1, \ldots, x_n)$: <prog>	Définition

Opération	<prefix>	Transformation
Création d'une localisation	$init(l)$ {<prog>}	Extension
Migration	$go(l_1 \rightarrow l_2)$	Extension/Translation
Création d'un canal	$\nu(c)$	Extension
Acquisition d'un canal	$dock_l(c)$	Coloriage
Relâchement d'un canal	$undock_l\langle c\rangle$	Coloriage
Connexion	$link_l(c)$	Coloriage
Déconnexion	$unlink_l\langle c\rangle$	Coloriage
Émission	$c^+\langle x\rangle$	Extension/Remplissage
Réception	$c^-(x)$	Translation

Tableau 1. *Syntaxe des programmes* $cube$*-calcul*

2. Le $cube$-calcul, un langage pour les agents mobiles

Nous développons dans cette section la syntaxe et la sémantique informelle d'un langage de description de comportements d'agents mobiles, le $cube$-calcul. D'un point de vue purement syntaxique, le langage proposé peut être vu comme une extension, par quelques mots-clés seulement, du π-calcul.

La syntaxe BNF du $cube$-calcul est présentée sur le tableau 1. Les opérateurs de choix et de composition parallèle sont ceux du π-calcul. Des préfixes spécifiques sont introduits pour la communication et la mobilité. Nous indiquons, par anticipation sur la sémantique géométriques discutée en section 3, les catégories de transformation concernée par chacun de ces préfixes.

En matière d'expressivité, le $cube$-calcul retient *au moins* toute la puissance de son aîné le π-calcul, ou plutôt d'une variante localisée : communication, mobilité des liens et mobilité des agents sont présentes[1]. En complément du point de vue subjective généralement adopté et qui impose aux agents d'être « conscients » de leur capacités mobiles, nous illustrons le support objectif de la mobilité dans le $cube$-calcul. Nous discutons préalablement de la sémantique de communication du $cube$-calcul et ce qui en fait selon nous un langage « réaliste », c'est-à-dire un candidat viable pour une implémentation répartie efficace.

1. À priori, seule la mobilité synchrone des liens est nécessaire pour « égaler » le π-calcul en terme d'expressivité.

2.1. *Conflits de ressources*

Un grief récurrent concernant la possibilités d'implémenter le π-calcul (dans sa variante asynchrone [AMA 01]) en environnement distribué concerne les *conflits de ressources*. Les conflits se produisent lorsque deux agents distincts possèdent une référence vers le même canal de communication. Deux cas conflictuels peuvent se produire : (1) *conflit de lecture* ou (2) *conflit d'écriture* si les deux agents veulent respectivement lire ou écrire sur le même canal *au même moment*. Pour le cas de l'écriture, le terme suivant explicite la situation de conflit :

$$c^{+}_{l_1}\langle a\rangle \parallel c^{+}_{l_2}\langle b\rangle \parallel c^{-}_{l_3}(x).P(x)$$

Observé d'un point de vue distribué, le problème est flagrant : l_1 et l_2 peuvent émettre *au même moment*, sur le *même canal* c *deux valeurs différentes* a et b. Le join calculus [FOU 96] propose de *localiser* c en interdisant les conflits distribués que ce soit en lecture ou en écriture. Dans le $cube$-calcul, le choix est encore plus restrictif puisque la capacité d'écriture sur c, obtenue par le préfixe $dock(c)$, est *unique*. Cette propriété sera vérifiée *par construction* dans les espaces d'interaction (cf. section 3). Pour le cas de lecture, le terme caractéristique est le suivant :

$$c^{+}_{l_1}\langle a\rangle \parallel c^{-}_{l_2}(x).P(x) \parallel c^{-}_{l_3}(y).Q(y)$$

Ici, on se pose la question de qui de l_2 et l_3 doit recevoir la valeur a émise sur c par l_1. Il est encore possible de *localiser* le canal c comme dans le cas du join calcul mais nous y voyons une restriction quelque peu contre-intuitive. Ce que nous « lisons » dans cet exemple, c'est que l_1 désire de son côté communiquer une valeur que l_2 et l_3 sont tous deux susceptibles de recevoir. Une interprétation naturelle de ce terme consiste à recevoir l'information à la fois en l_2 et en l_3 (mais pas forcément en même temps bien sûr puisque la sémantique de communication est asynchrone). Au delà de la résolution des conflits, ce choix du *multipoint* par défaut s'accompagne de nombreuses propriétés intéressantes [ENE 99] [PES 02].

2.2. *Asynchronisme et causalité*

Les langages mobiles proposés dans la littérature prennent peu souvent en compte les propriétés d'ordre, de *causalité*, dans la sémantique de communication. Prenons l'exemple d'un système transactionnel composés d'agents clients et serveurs de transaction. Du point de vue d'un client situé en l, une transaction est une séquence de requêtes de la forme :

$$Client(tr, l) : tr^{+}_{l}\langle open\rangle.tr^{+}_{l}\langle req_1\rangle.tr^{+}_{l}\langle req_2\rangle \ldots tr^{+}_{l}\langle req_n\rangle.tr^{+}_{l}\langle close\rangle$$

Le canal tr est utilisé pour la communication client/serveur. Il est clair ici que la relation d'*ordre causal* entre les requêtes doit être préservée d'une façon ou d'une autre du côté du serveur. Ce dernier peut être vu comme une composition d'agents communiquants que nous pouvons abstraire de la façon suivante :

$$Server(tr, m) : tr_m^-(x).Proxy(s) \parallel Serve_1 \parallel \ldots \parallel Serve_n$$

La partie $Proxy$ se charge de communiquer avec le client et les agents $Serve_i$ prennent en charge l'exécution concurrente des transactions. Si l'on prend l'exemple du π-calcul asynchrone [AMA 01], la préservation de l'ordre causal dans la transmission des requêtes du client est difficile à caractériser. En effet, puisque les préfixes d'émission ne peuvent disposer de continuations, l'ordre causal doit être *encodé* par des processus concurrents. Il faudra par exemple écrire un expression de la forme :

$$run_1^-.tr_l^+\langle req_1\rangle \parallel run_2^-.tr_l^+\langle req_2\rangle \parallel \ldots \parallel run_1^+ \parallel tr_m^-(req_1).run_2^+ \parallel \ldots$$

Nous devons introduire ici du parallélisme et de la communication alors que l'objectif est d'encoder une séquence locale et déterministe. Pour obtenir un encodage plus « naturel » de cette séquence, nous pouvons considérer que le canal tr achemine les messages selon une sémantique *fifo* [TEL 00]. Dans ce cas, le programme de départ, plus intuitif, devient valide sans pour autant dénoter une exécution synchrone. Ce choix est celui du $cube$-calcul qui, contrairement à la plupart des variantes du π-calcul asynchrone, permet aux préfixes d'émissions de disposer de continuations.

2.3. *La mobilité dans le* $cube$*-calcul*

Le $cube$-calcul supporte deux formes de mobilité : *mobilité des agents* et *mobilité des liens* ou canaux de communication. La mobilité des agents, ou *migration*, correspond au changement de localisation (physique ou logique) d'un agent pendant son exécution. Il s'agit de la forme la plus courante (mais pas la plus simple à mettre en œuvre [SEK 99]) de mobilité. Le π-calcul introduit une forme plus virtuelle de mobilité : la mobilité des liens. Pour saisir l'importance de ce concept, il faut voir dans les liens de communications l'explicitation de l'*environnement dynamique* des agents. Lors de la migration d'un agent, l'environnement n'est pas perturbé puisque seule la localisation de l'agent change. Sans modification environnementale, la notion même de mobilité perd ainsi une bonne partie de son intérêt. Pour illustrer ces concepts, plaçons nous dans le cadre de l'exemple décrit en figure 1 page 36. Trois types d'agents sont mis en jeu : client, serveur et interface graphique.

2.3.1. *Mobilité objective des agents*

La migration de l'application cliente (par exemple en réaction à un déplacement de son utilisateur) correspond au mouvement ①. Dans le $cube$-calcul, il suffit d'une instruction, $go(l_1 \rightarrow l_2)$, pour initier ce mouvement. Il est notable que ce changement de localisation est *objectif* : il peut être invoqué depuis n'importe quelle localisation du système. Considérons l'exécution suivante :

$$go_{l_3}(l_1 \rightarrow l_2).P_{l_3} \parallel c_{l_1}^+\langle a\rangle.Q_{l_1} \rightarrow P_{l_3} \parallel c_{l_2}^+\langle a\rangle.Q_{l_2}$$

Ici, l'agent localisé en l_1, *actif* car en cours d'émission sur le canal c, voit sa localisation modifiée par l'agent l_3. Ce type d'expression et de sémantique de réduction

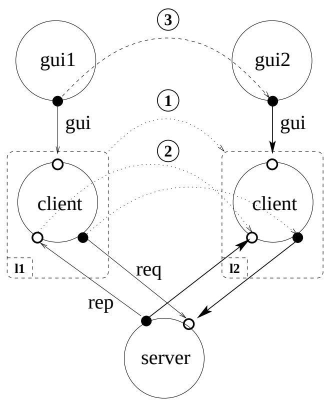


Figure 1. *Mobilité des agents ① et des canaux de communication ②③*

n'est généralement pas permise dans les algèbres mobiles. Le déplacement d'un agent actif est pourtant tout à fait utile et implémentable en pratique [MAS 04].

2.3.2. *Mobilité des canaux de communication*

Comme le montre la figure 1, le déplacement de l'agent client induit un deuxième type de modification : le mouvement de tous les canaux de communication rattachés au composant mobile (marque ②). En revanche, le déplacement du canal gui qui relie le client à son interface graphique (marque ③) n'est pas automatique. Par défaut, le mouvement du client préserve la connexion avec l'interface graphique de départ gui_1. Mais si l'utilisateur se déplace, il est clair que son interface graphique doit également, d'une façon ou d'une autre, changer. C'est ce que modélise le mouvement ③ du canal gui vers l'agent gui_2. Le π-calcul s'avère très expressif pour spécifier ce type de reconfiguration. Supposons que le serveur soit en charge de la mobilité, via des canaux de contrôle c_1 (contrôle du client) et c_2 (contrôle de l'interface graphique). Ce que nous voulons, c'est connecter gui_2 au client l_2 pour qu'ils puissent communiquer. En π-calcul (enrichi de localisations[2]), nous écrirons par exemple :

$$c1^{+}_{l_3}\langle gui\rangle.c2^{+}_{l_3}\langle gui\rangle \parallel c1^{-}_{gui_2}(x).x^{+}_{gui_2}\langle e\rangle \parallel c2^{-}_{l_2}(y).y^{-}_{l_2}(z).P_{l_2}(z)$$

La réduction de ce programme dans la sémantique standard du π-calcul conduira à la transmission de e (évènement d'interface graphique) sur le canal gui, de l'agent gui_2 vers le client l_2. En π-calcul, communication et modifications topologiques sont unifiées. Si l'on désire raisonner sur les modifications topologiques, alors l'abstraction proposées est adéquate mais si l'on est concerné par l'implémentation pratique de cet exemple, des mécanismes non triviaux sont révélés. Tout d'abord, on peut distinguer deux problèmes bien différents dans l'exemple proposé : (1) l'acheminement de la

2. Les localisations ne sont pas nécessaires pour réduire le terme proposé mais elles mettent en relief la nature distribuée du système.

$P \equiv P\{y/x\}$ avec x lié et y non-libre dans P	$alpha$-convertibilité
$P \parallel Q \equiv Q \parallel P$	commutativité de $\parallel$
$P \parallel 0 \equiv P$	élément neutre de $\parallel$
$P + Q \equiv Q + P$	commutativité de $+$
$P + 0 \equiv P$	élément neutre de $+$
$D(\tilde{y}) \equiv P\{\tilde{y}/\tilde{x}\}$ si $D(\tilde{x}) : P$	Récursion

Tableau 2. *Relation de congruence structurelle*

Opérateurs :

$$\frac{\Delta \vdash P \rightarrow \Delta' \vdash P'}{\Delta \vdash P \parallel Q \rightarrow \Delta' \vdash P' \parallel Q}\ (O1) \qquad \frac{\Delta \vdash P \rightarrow \Delta' \vdash P'}{\Delta \vdash P + Q \rightarrow \Delta' \vdash P'}\ (O2)$$

$$\frac{P \equiv P' \quad \Delta \vdash P \rightarrow \Delta' \vdash Q \quad Q \equiv Q'}{\Delta \vdash P' \rightarrow \Delta' \vdash Q'}\ (O3)$$

Tableau 3. *Sémantique des opérateurs du langage*

référence du canal gui vers les agents gui_2 et l_2 et (2) l’établissement dynamique du lien reliant ces deux agents. Le premier cas correspond à une communication de valeur, qui pourra s’effectuer de façon asynchrone (et donc efficace) dans le cadre d’un système distribué réel. Le deuxième cas correspond à un cas inévitablement synchrone (et plus coûteux) de reconfiguration topologique. Dans le $cube$-calcul, communications et modifications topologiques sont clairement séparées. Le terme correspondant est le suivant :

$$c1^+_{l_1}\langle d\rangle.c2^+_{l_1}\langle d\rangle \parallel c1^-_{l_2}(x).dock_{l_2}(x).x^+_{l_2}\langle e\rangle \parallel c2^-_{l_3}(y).link_{l_3}(y).y^-_{l_3}(z).P(z)$$

Ce terme est très proche de la version précédente en π-calcul. Nous voyons ainsi que le passage du monde « idéal » du pi-calcul au monde « réaliste » du $cube$-calcul est très simple.

3. Les espaces d’interaction

Le $cube$-calcul, introduit dans la section précédente, bien qu’inspiré et proche du π-calcul dans les principes, s’en éloigne inexorablement sur le versant sémantique. Nous développons dans cette section une sémantique opérationnelle structurelle pour ce langage. En ce qui concerne les rapprochement avec les algèbres de processus, nous introduisons une relation de congruence structurelle (cf. tableau 2) ainsi qu’une sémantique standard pour les opérateurs du langage (cf. tableau 3). Les préfixes du $cube$-calcul concernant la communication et la mobilité sont traités dans le cadre du modèle des espaces d’interaction. Ces espaces reposent sur une intuition géométrique. Nous proposons de modéliser comme de simples transformations géométriques les opérations qui nous semblent fondamentales pour l’étude de langages distribués et mobiles.

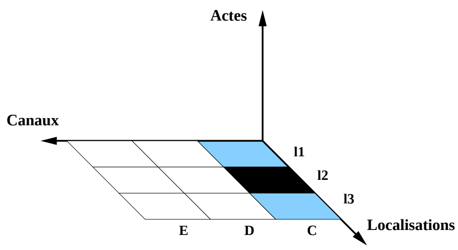


Figure 2. *La géométrie des espaces d'interaction*

Un espace d'interaction, noté Δ, se représente selon la figure 2. L'état d'un programme P dans cet espace se note $\Delta \vdash P$. La géométrie d'un espace d'interaction comporte trois dimensions :

1) la dimension $\mathcal{L}$ des *localisations*,

2) la dimension $\mathcal{C}$ des *canaux de communication*, et

3) la dimension $\mathcal{A}$ des *actes* (état des canaux).

Les transformations définies sur cette géométrie se regroupent en quatre classes :

– **expansion** $\mathcal{E}$,

– **translation** $\mathcal{T}_{\overrightarrow{v}}$ (de vecteur $\overrightarrow{v}$),

– **coloriage** $\mathcal{C}_c$ (de couleur c), et

– **remplissage** $\mathcal{F}_v$ (de valeur v).

Il est possible de restreindre la portée d'une transformation en la confinant, par une **projection**, à un sous-espace. Si Φ est une transformation alors $\Phi\Delta$ transforme tout l'espace. Si δ est une projection de Δ alors $\Phi\delta$ ne transforme que le sous-espace δ de Δ. La projection de l'espace vers un sous-espace se note $\Delta|_{\lambda,\gamma,\alpha}$ où λ, γ et α sont respectivement les sous-espaces de localisations, canaux et actes. L'espace dans son intégralité est $\Delta|_{\mathcal{L},\mathcal{C},\mathcal{A}}$. La projection de l'espace sur un canal $c \in \mathcal{C}$ se note $\Delta|_{\mathcal{L},c,\mathcal{A}}$. Il est également possible de ne pas considérer une ou plusieurs dimensions pour la projection. Par exemple, la dimension des localisations seule est notée $\Delta|_{\mathcal{L},\emptyset,\emptyset}$. Ceci nous permet de définir l'espace vide : $\Delta|_{\emptyset,\emptyset,\emptyset} = \emptyset$.

Les opérations fondamentales des systèmes d'agents distribués et mobiles qui nous intéressent seront définies par composition de ces transformations géométriques élémentaires.

3.1. *Topologie explicite*

Avant de pouvoir communiquer, les espaces d'interaction imposent de définir précisément la *topologie* du système. Il s'agit bien sûr de positionner les agents dans l'espace, mais également d'expliciter la géométrie des canaux de communications qui

Expansions :

$$\frac{}{\Delta \vdash init_{l_1}(l_2)\,\{Q\}\,.P_{l_1} \rightarrow \textbf{\textit{newloc}}(\Delta) \vdash P_{l_1} \parallel Q_{l_2}\{\textbf{\textit{locref}}(\Delta)/l_2\}}\ (E1)$$

$$\frac{}{\Delta \vdash \nu(c).P \rightarrow \textbf{\textit{newchan}}(\Delta) \vdash P\{\textbf{\textit{chanref}}(\Delta)/c\}}\ (E2)$$

Coloriages :

$$\frac{owner(\Delta, c) = \emptyset}{\Delta \vdash dock_l(c).P \rightarrow \textbf{\textit{set}}(\Delta, l, c, owner) \vdash P}\ (C1)$$

$$\frac{owner(\Delta, c) = \{l\}}{\Delta \vdash undock_l\langle c\rangle.P \rightarrow \textbf{\textit{set}}(\Delta, l, c, empty) \vdash P}\ (C2)$$

$$\frac{owner(\Delta, c) \neq \{l\}}{\Delta \vdash link_l(c).P \rightarrow \textbf{\textit{set}}(\Delta, l, c, connected) \vdash P}\ (C3)$$

$$\frac{l \in connected(\Delta, c)}{\Delta \vdash unlink_l\langle c\rangle.P \rightarrow \textbf{\textit{set}}(\Delta, l, c, empty) \vdash P}\ (C4)$$

Tableau 4. *Gestion de la topologie*

relient les agents entre eux. Les règles de réduction concernées sont définies sur le tableau 4. Créer une localisation pour un agent (règle $E1$) se fait par une expansion unitaire de la dimension $\mathcal{L}$ des localisations. La « valeur » de la localisation créée correspond au cardinal de la dimension des localisations. De façon plus formelle, nous définissons :

$$\textbf{\textit{newloc}}(\Delta) \doteq \mathcal{E}\Delta|_{\mathcal{L},\emptyset,\emptyset} \qquad \textbf{\textit{locref}}(\Delta) \doteq card(\Delta|_{\mathcal{L},\emptyset,\emptyset})$$

Remarquablement, la référence retournée par $\textbf{\textit{locref}}(\Delta)$ est *unique par construction*. Nous avons en effet : $\textbf{\textit{locref}}(newloc(\Delta)) = \textbf{\textit{locref}}(\Delta) + 1$. Les canaux de communication se créent de façon similaire (règle $(E2)$). Nous définissons ainsi :

$$\textbf{\textit{newchan}}(\Delta) \doteq \mathcal{E}\Delta|_{\emptyset,\mathcal{C},\emptyset} \qquad \textbf{\textit{chanref}}(\Delta) \doteq card(\Delta|_{\emptyset,\mathcal{C},\emptyset})$$

Encore une fois, l'identification unique d'un canal est vérifiée par construction. Pour pouvoir envoyer de l'information, un agent doit devenir *propriétaire* d'un canal de communication. Ce propriétaire est globalement unique pour chaque canal. Symétriquement, un agent désirant recevoir des informations sur un canal donné doit être *connecté* à ce canal. Nous introduisons la notion de *couleur* pour expliciter ces informations dans la géométrie proposée (règles $C1$, $C2$, $C3$ et $C4$ de la sémantique opérationnelle). Sur la figure 2 page 38, l'agent l_2 est défini comme propriétaire du canal c. La couleur employée est notée $owner$, elle est représentée en noir sur la figure. De même, les agents l_1 et l_3 sont connectés au canal c, c'est-à-dire *coloriés* avec la couleur $connected$ (grisé). Pour colorier l'intersection d'une localisation et d'un canal, la transformation employée est : $\textbf{\textit{set}}(\Delta, l, c, color) \doteq \mathcal{C}_{color}\Delta|_{l,c,\mathcal{A}}$

Remplissage :

$$\frac{owner(\Delta, c) = \{l\}}{\Delta \vdash c_l^+\langle e\rangle.P \rightarrow \textbf{\textit{send}}(\Delta, c, l, e) \vdash P}\ (R)$$

Translations :

$$\frac{l \in connected(\Delta, c)}{\Delta \vdash c_l^-(x).P \rightarrow \textbf{\textit{receive}}(\Delta, l, c) \vdash P\{\textbf{\textit{fetch}}(\Delta, l, c)/x\}}\ (T1)$$

$$\frac{}{\Delta \vdash go(l_1 \rightarrow l_2).P \rightarrow \textbf{\textit{move}}(\Delta, l_1) \vdash P}\ (T2)$$

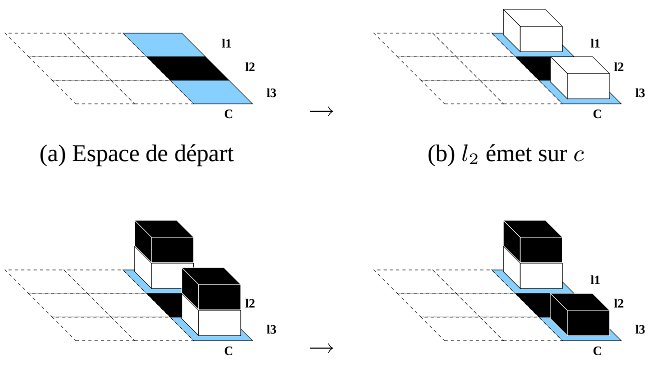
Tableau 5. *Sémantique pour l'émission* (R)*, la réception* $(T1)$ *et la migration* $(T2)$

Pour obtenir l'espace de la figure 2, les transformations considérées sont donc :

$\textbf{\textit{set}}(\Delta, l_2, c, owner)$ $\quad$ $\textbf{\textit{set}}(\Delta, l_1, c, connected)$ $\quad$ $\textbf{\textit{set}}(\Delta, l_3, c, connected)$

Nous introduisons également la notation $color(\Delta)$ pour extraire le sous-espace d'une couleur donnée. Ainsi, nous notons $connected(\Delta)$ le sous-espace de Δ de couleur $connected$. Par combinaison avec une projection de l'espace, nous pouvons par exemple obtenir l'ensemble des agents connectés à un canal donné. Dans la configuration de la figure 2, nous aurons donc : $connected(\Delta|_{\mathcal{L},c,\emptyset}) = \{l_1, l_2\}$, c'est-à-dire les localisations des agents connectés au canal c. Pour simplifier les notations, nous introduisons l'abréviation : $color(\Delta, c) \triangleq color(\Delta|_{\mathcal{L},c,\emptyset})$.

3.2. *La communication*



(a) Espace de départ

(b) l_2 émet sur c

(c) l_2 émet sur c

(d) l_3 reçoit sur c

Figure 3. *La communication dans les espaces d'interaction*

L'émission d'information par un agent dans un espace d'interaction correspond au « remplissage » de la dimension des actes (règle R de la sémantique opérationnelle, définie sur le tableau 5). Sur la figure 3(b) en page 40, l'agent l_2 émet une information (représentée par un cube blanc) sur le canal c dont il est le propriétaire. La transformation mise en jeu, graphiquement très simple, est de notation relativement complexe. Dans un premier temps, nous devons étendre l'espace dans la dimension des actes pour tout agent connecté au canal c, nous l'écrivons :

$$\textbf{\textit{newact}}(\Delta, c) \mathrel{\hat{=}} \mathcal{E}connected(\Delta|_{\mathcal{L},c,\mathcal{A}})|_{\emptyset,\emptyset,\mathcal{A}}$$

Dans un deuxième temps, nous pouvons remplir l'espace par la valeur e souhaitée :

$$\textbf{\textit{mcast}}(\Delta, c, e) \mathrel{\hat{=}} \mathcal{F}_e\Delta|_{\mathcal{L},c,card(\mathcal{A})}$$

L'émission correspond à la composition de ces deux opérations :

$$\textbf{\textit{send}}(\Delta, c, e) \mathrel{\hat{=}} \textbf{\textit{mcast}}(\textbf{\textit{newact}}(\Delta, c), c, e)$$

Sur la figure 3(a), l'agent l_2 émet une information “cube noir” sur le canal c. Comme les agents l_1 et l_3 sont tous deux connectés à c, l'information est copiée pour chacun d'entre eux. En figure 3(d), nous représentons l'opération réception par l'agent l_1. La valeur à recevoir correspond au point de l'espace de coordonnées $(l_1, c, 1)$, c'est-à-dire le premier (ou plus ancien) acte disponible. La réception proprement dite (règle $T1$ de la sémantique) correspond à une « simple » opération de translation unitaire dans la dimension des actes, à l'intersection du canal et de la localisation considérés. Dans notre notation, nous définissons :

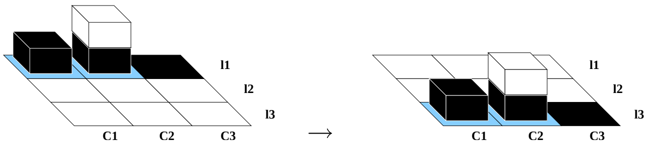
$$\textbf{\textit{receive}}(\Delta, l, c) \mathrel{\hat{=}} \mathcal{T}_{\overrightarrow{(0,0,-1)}}\Delta|_{l,c,\mathcal{A}} \qquad \textbf{\textit{fetch}}(\Delta, l, c) \mathrel{\hat{=}} \Delta|_{l,c,1}$$

La sémantique de communication *fiable*, *asynchrone*, *multipoint* et *fifo* du $cube$-calcul est donc ici proposée par défaut. Ces choix, motivés dans la section 2, ne sont pas figés dans le modèle. Nous pouvons par exemple modéliser les échanges *synchrones* en réduisant la capacité de mémorisation des canaux à une seule information[3]. Nous obtenons en quelque sorte un π-calcul « graphique ».

3.3. *La mobilité*

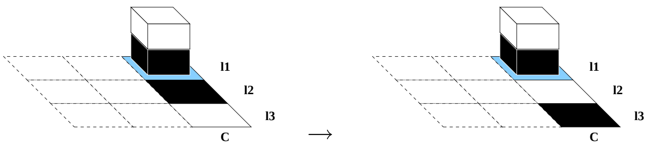
Les deux formes de mobilité qui nous intéressent, *mobilité des agents* et *mobilité des canaux*, sont également caractérisables précisément dans les espaces d'interaction. Le déplacement d'un agent dans un espace d'interaction est une opération simple de translation, combinée avec une expansion de la dimension des localisations (règle $T2$ de la sémantique). Par exemple, sur la figure 4(a), un agent se trouve sur la localisation l_1. Sur la figure 4(b), ce même agent a *migré* vers une nouvelle location l_3. La translation de l'espace se note ainsi :

3. Dans l'exemple précédent, il faudrait donc attendre avant de pouvoir émettre le cube noir qui ne peut être « placé » au-dessus du cube blanc avant que ce dernier soit consommé.



(a) Avant migration (b) Migration de l_1

Figure 4. *Mobilité des Agents (Migration)*



(a) Avant Mouvement (b) Mouvement de c

Figure 5. *Mobilité des canaux de communication*

$$\textbf{\textit{migrate}}(\Delta, l_1) \doteq \mathcal{T}_{\overrightarrow{(l_2, 0, 0)}} \Delta|_{l_1, \mathcal{C}, \mathcal{A}} \text{ avec } l_2 = card(\Delta|_{\mathcal{L}, \emptyset, \emptyset})$$

Cette définition, quoique concise, sous-entend la préservation de l'ensemble des informations relatives à l_1, en particulier l'état des canaux auxquels il est connecté (cf. [MAS 04] pour les mécanismes opérationnels mis en jeu). L'opération complète de déplacement d'agent se note : $\textbf{\textit{move}}(\Delta, l) \doteq \textbf{\textit{migrate}}(\textbf{\textit{newloc}}(\Delta), l)$

Les mouvements des canaux de communication sont capturés par les opérations de modifications topologiques introduites précédemment (cf. règles $C1$ et $C2$, tableau 4). Pour déplacer un lien, il suffit de changer dynamiquement son propriétaire. Sur la figure 5(a), le canal c a pour propriétaire l'agent l_2. Le mouvement du lien « vers » la location l_3 correspond au changement de propriétaire pour le canal. Ceci se déroule en deux étapes : (1) libération du canal par l_2 ($\textbf{\textit{set}}(\Delta, c, l_2, empty)$) puis acquisition par l_3 ($\textbf{\textit{set}}(\Delta, c, l_3, owner)$).

3.4. *Une équivalence pour le $cube$-calcul*

Une relation d'équivalence pour les programmes du $cube$-calcul s'obtient simplement à partir du système de réduction proposé. Nous dirons que P *est équivalent à* Q, et noterons, $\Delta \vdash P \approx \Delta \vdash Q$, si et seulement si, symétriquement :

si $\Delta \vdash P \to^* f(\Delta) \vdash P'$ alors il existe un Q' tel que $\Delta \vdash Q \to^* g(\Delta) \vdash Q'$ et $f(\Delta) = g(\Delta) = \Delta'$ et $\Delta' \vdash P' \approx \Delta' \vdash Q'$

Cette relation d'équivalence linéaire nous ouvre d'ores et déjà les portes de la vérification de modèle pour les programmes du $cube$-calcul. Elle nous permet également de trouver des équivalences syntaxiques intéressantes. Nous pouvons par exemple montrer fort simplement[4] que les émissions sur des canaux distincts sont commutatives. C'est-à-dire que si $c_1 \neq c_2$ alors $c_1{}^+\langle a\rangle.c_2{}^+\langle b\rangle \approx c_2{}^+\langle b\rangle.c_1{}^+\langle a\rangle$

4. Travaux connexes

D'un certain point de vue, notre proposition peut être vue comme une variation supplémentaire sur le thème du π-calcul [MIL 99]. De ce langage nous retenons principalement la simplicité et l'expressivité du concept de *mobilité des liens* pour discuter de la caractérisation des changements environnementaux se produisant pendant l'exécution des applications. Malgré son interprétation asynchrone, le $cube$-calcul est syntaxiquement plus proche de la version standard du π-calcul que de sa version asynchrone [AMA 01]. Ce dernier interdit en effet, de façon selon nous plutôt contre-intuitive, les continuations de préfixes d'émissions. Les espaces d'interaction, par leur caractérisation simple des canaux *fifo*, éliminent cette contrainte.

L'immersion des concepts du π-calcul dans l'environnement distribué des espaces d'interaction ne se résume pas à l'explicitation de localisations. Comme dans le modèle des acteurs [AGH 97], une séparation claire entre système (espace d'interaction) et programme (expression du $cube$-calcul) est proposée. De ce point de vue, nous pouvons voir les espaces d'interaction comme un enrichissement des configurations d'acteur par l'explicitation des liens de communication (et la sémantique associée) et la mobilité. Selon nous, la distribution passe également par la distinction entre sources et récepteurs d'information. On retrouve cette *polarisation* [ODE 95] dans la plupart des sémantiques appliqués de langages mobiles comme le $join$-calcul distribué [FOU 96]. Dans ce langage et dans la plupart des variantes distribuées du π-calcul, seule la capacité d'émission sur un canal est transmissible. On parle dans ce cas de π-calcul *local* [MER 98]. Cette restriction n'est pas présente dans les espaces d'interaction, la capacité de lecture étant également transmissible (connexion de canal). Enfin, les possibilités de communications multipoint sont explorées dans le pi-calcul *broadcast* [ENE 99]. Les récepteurs multiples sur un canal donné sont identifiés dans la syntaxe des programmes. Une relation supplémentaire est introduite pour se « débarrasser » des communications non-désirées (*discard*). Dans les espaces d'interaction, le point de vue est exactement inverse puisque les connexions sont établies de façon « constructive », via les préfixes spécifiques de connexion et de de déconnexion. De plus, il est possible de traiter le cas unipoint en garantissant l'unicité de la connexion sur un canal, il ne s'agit donc pas de proposer le multipoint comme mode de communication *incontournable*.

4. Il existe une preuve de cette propriété dans notre implémentation fonctionnelle des espaces d'interaction en ACL2 (cf. `http://web.yl.is.s.u-tokyo.ac.jp/~pesch/is`).

La mobilité objective est à la fois motivée et critiquée par les *ambients mobiles* [CAR 00] et nomadic pict [UNY 01] dans lesquels les primitives subjectives et objectives ne sont pas inter-définissables. Dans les espaces d'interaction , la mobilité subjective y est vue comme un cas particulier du cas objectif. Ainsi, un agent peut objectivement se déplacer lui-même. Une perspective de recherche intéressante concerne, dans l'intuition des ambients, la hiérarchisation de la dimension des localisations (pour l'instant linéaire). On pourrait voir la géométrie résultant comme un emboîtement d'espaces d'interaction.

5. Conclusion et travaux futurs

Les espaces d'interaction caractérisent les interactions complexes se produisant dans les systèmes d'agents mobiles sous la forme de transformations géométriques. Le point de vue proposé est objectif, il permet d'observer le système dans son intégralité. Prenons l'exemple des systèmes de type client/serveur. Il nous semble difficile de saisir l'essence des interactions mises en jeu par une observation subjective, depuis chaque client ou serveur du système. En revanche, la géométrie obtenue dans les espaces d'interaction est caractérisable simplement. Ce sont des « points qui clignotent », géométries primitives pour les interactions synchrones. Autre exemple, un *verrou* de communication correspond à un point fixe qui s'éloigne inexorablement de la *surface active* de l'espace (le lieu des communications en cours). Une structure « en spirale » pourra identifier des problèmes de *vivacité*, etc. Nous n'en sommes bien sûr qu'aux investigations préliminaires de ces « mesures géométriques » mais il s'agit nous le pensons d'un axe de recherche assez prometteur.

Sur ces fondations géométriques, nous construisons un langage qui permet la description des comportements d'agent. Ce $cube$-calcul retient l'ensemble des propriétés du π-calcul dont il s'inspire et l'étend par les opérations spécifiques des espaces d'interaction : mobilité objective des liens et des agents, communications asynchrones, multipoint et ordonnées, etc. Malgré l'exposé d'une relation d'équivalence sur les programmes du $cube$-calcul, il est clair que le versant formel du langage est moins développé que d'autres approches appliquées du π-calcul. Il nous semble cependant possible de fournir une sémantique compositionnelle en capturant, probablement par l'intermédiaire de système de transitions labellées, l'environnement des agents. Pour autant, les raisonnements « de niveau langage » sont d'ores et déjà possibles dans notre approche grâce aux nombreuses propriétés capturées directement par les espaces d'interaction.

Dans un travail séparé [MAS 04], nous implémentons une version enrichie du $cube$-calcul dans le cadre de notre intergiciel Comet. Cette implémentation préliminaire permet de faire le lien avec la pratique et de confirmer le positionnement « réaliste » de notre approche.

6. Bibliographie

[AGH 97] AGHA G., MASON I. A., SMITH S. F., TALCOTT C. L., « A foundation for actor computation », *Journal of Functional Programming*, vol. 7, n° 1, 1997, p. 1–69.

[AMA 01] AMADIO R., CASTELLANI I., SANGIORGI D., « On bisimulations for the asynchronous pi-calculus », *CONCUR'96*, vol. 1119 de *LNCS*, Springer Verlag, 2001.

[CAR 00] CARDELLI L., GORDON A. D., « Mobile Ambients », *Theoretical Computer Science*, vol. 240, n° 1, 2000, p. 177–213, Cambridge University Press.

[ENE 99] ENE C., MUNTEAN T., « Expressiveness of point-to-point versus broadcast communications », *Symposium : Fundamentals of Computation Theory*, vol. LNCS 1684, Springer Verlag, 1999.

[FOU 96] FOURNET C., GONTHIER G., LÉVY J.-J., MARANGET L., RÉMY D., « A Calculus of Mobile Agents », *7th International Conference on Concurrency Theory (CONCUR'96)*, vol. 1119 de *LNCS*, Pisa, Italy, août 26-29 1996, Springer, p. 406–421.

[MAS 04] MASUYAMA T., PESCHANSKI F., OYAMA Y., YONEZAWA A., « MobileScope : a Language with Objective Mobility », *Proceedings of MDC'04, ICDCS 2004 Workshops*, IEEE, 2004, à paraitre.

[MER 98] MERRO M., SANGIORGI D., « On asynchrony in name-passing calculi », *Proc. 25th ICALP*, vol. 1443 de *LNCS*, Springer Verlag, 1998.

[MIL 99] MILNER R., *Communicating and Mobible Systems : The π-Calculus*, Cambridge University Press, 1999.

[ODE 95] ODERSKY M., « Polarized Name Passing », *Proc. FST & TCS*, LNCS, Springer Verlag, décembre 1995.

[PES 02] PESCHANSKI F., « A Versatile Event-based Communication Model for Generic Distributed Interactions », *Proceedings of DEBS'02 (ICDCS International Workshop on Distributed Event-based Systems)*, IEEE, Juillet 2002.

[SEK 99] SEKIGUCHI T., MASUHARA H., YONEZAWA A., « A Simple Extension of Java Language for Controllable Transparent Migration and its Portable Implementation », *Coordination'99*, vol. LNCS 1594, Springer-verlag, 1999.

[TEL 00] TEL G., *Introduction to Distributed Algorithms*, Cambridge University Press, 2000.

[UNY 01] UNYPOTH A., SEWELL P., « Nomadic Pict : Correct Communication Infrastructure for Mobile Computation », *POPL 2001. ACM Sigplan Notices*, vol. 36, March 2001.

Annexe : implantation fonctionnelle des espaces d'interaction en ACL2

Par manque de place, cette annexe est uniquement disponible en ligne :
`http://web.yl.is.s.u-tokyo.ac.jp/~pesch/is/medias/lmo2004annexe.ps`