On Mobility and Communication

Wenjie Du [a], Yuxin Deng [b]

[a] *Shanghai Normal University, China*
wenjiedu@shnu.edu.cn

[b] *Shanghai Jiao Tong University, China*
yuxindeng@sjtu.edu.cn

Abstract

By imposing three semantic constraints on the π-calculus equipped with delayed input, we obtain a variant of the π-calculus. We prove that Hennessy and Riely's distributed π-calculus can be encoded into this new model, satisfying a full abstraction theorem, which means that one-dimensional mobility can be characterized by communication. We also discuss the difficulties of expressing two-dimensional mobility, the mechanism used by ambient calculi.

Key words: mobility, communication, distributed π-calculus, delayed input

1 Introduction

In the community of process algebra, there exist different kinds of strategies for controlling resource access, which entail different models for describing concurrency and mobility. For example, in the standard π-calculus [8], the restriction operator can be considered as a tool of limiting names for local use. In distributed π-calculus (Dπ) [6], location is introduced to realize this function, since processes in different locations are not able to interact with each other directly. In mobile ambients [4], every ambient has a boundary which distinguishes interior processes from exterior ones.

But all those controlled resources are not isolated. To describe mobile systems, i.e., systems with a dynamically changing linkage topology, each model must provide a mechanism to allow resources to extrude their original scope so as to be used by processes outside the scope. For instance, in the π-calculus, the following reduction relation holds.

$$(1) \qquad (\nu b)(\bar{a}b.P|Q)|a(x).R \longrightarrow (\nu b)(P|Q|R\{b/x\})$$

The private name b, which is previously shared by $\bar{a}b.P$ and Q, is transmitted along a. As a consequence, the scope of b is extended to $R\{b/x\}$. Consider the following example from Dπ:

(2) $l[\texttt{go } k.\bar{a}b.P|Q]|k[a(x).R] \longrightarrow l[Q]|k[\bar{a}b.P]|k[a(x).R]$

(3) $\longrightarrow l[Q]|k[P|R\{b/x\}]$

After the first step of reduction, the agent $\bar{a}b.P$ moves from one location to another, say from l to k. Then it can communicate with $a(x).R$ which stays in location k. In mobile ambients, mobility paves the way for one ambient to enter the range of another. A typical example is given below:

(4) $n[\texttt{in } m.\langle b\rangle.P|Q]|m[\texttt{open } n.(x).R] \longrightarrow m[n[\langle b\rangle.P|Q]|\texttt{open } n.(x).R]$

(5) $\longrightarrow m[\langle b\rangle.P|Q|(x).R]$

(6) $\longrightarrow m[P|Q|R\{b/x\}]$

To send the name b to process R, ambient n has to enter the domain of m, be opened and then communicate with $(x).R$.

In example (1), only communication happens. In the next two examples, both mobility and communication take place. A careful observation reveals that (2) and (4) use different sorts of mobility, which we call one-dimensional mobility and two-dimensional mobility respectively. In (2), all locations are always located on the same level, no location nesting being permitted, while in (4), ambient n becomes a subambient of m after its movement.

In this paper, we will show that one-dimensional mobility can be characterized by communication if we use the semantics of delayed input [10,7]. Such a delayed input prefix, written $a(x) : P$, allows the continuation P to evolve underneath the input guard, except for observable actions related to x. Hence, it allows more parallelism in processes. However, the sequentialization forced by the ordinary prefix is useful for expressing the behaviors of many concurrent systems. So in this paper, instead of replacing one with the other, we keep both operators, with the requirement that the delayed input only possesses a part of its functions under the ordinary meaning. First, we give a restricted form of the π-calculus with delayed input (π_{DI}), called π_{rdi}. It is obtained by allowing only a portion of the legal operations in ordinary semantics of delayed input. In other words, if we use $\mathcal{S}_1$ and $\mathcal{S}_2$ to stand respectively for all legal operations in π_{rdi} and π_{DI}, then $\mathcal{S}_1$ is actually a subset of $\mathcal{S}_2$. Then we give an encoding of Dπ into π_{rdi}. So the reductions in example (2) will be simulated as follows:

(7) $l : \bar{l}.k : \bar{a}b.\bar{k}.\langle\!\langle P\rangle\!\rangle^k|\langle\!\langle Q\rangle\!\rangle^l|k : a(x).\bar{k}.\langle\!\langle R\rangle\!\rangle^k$

(8) $\xrightarrow{\tau} k : \bar{a}b.\bar{k}.\langle\!\langle P\rangle\!\rangle^k|\langle\!\langle Q\rangle\!\rangle^l|k : a(x).\bar{k}.\langle\!\langle R\rangle\!\rangle^k$

(9) $\xrightarrow{\tau} k : \bar{k}.\langle\!\langle P\rangle\!\rangle^k|\langle\!\langle Q\rangle\!\rangle^l|k : \bar{k}.\langle\!\langle R\rangle\!\rangle^k\{b/x\}$

(10) $\xrightarrow{\tau}{}^2 \langle\!\langle P\rangle\!\rangle^k|\langle\!\langle Q\rangle\!\rangle^l|\langle\!\langle R\rangle\!\rangle^k\{b/x\}$

(11) $\equiv \langle\!\langle Q\rangle\!\rangle^l|\langle\!\langle P|R\{b/x\}\rangle\!\rangle^k$

where l and $\bar{k}$ are abbreviations of $l(y)$ and $\bar{k}(z)$ respectively, for some y and

z, since we are not concerned with the names transmitted. We prove that our encoding meets the correctness criteria: operational correspondence and full abstraction. At last, we give a brief discussion about the difficulties of simulating two-dimensional mobility by communication.

In Section 2, we review Hennessy and Riely's Dπ and give a labelled transition system which is new because [6] only provides a reduction semantics. In Section 3, we present the syntax and semantics of π_{rdi}. In Section 4, we report the encoding of Dπ into π_{rdi}, which is fully abstract. In Section 5, we discuss two-dimensional mobility. Finally in Section 6, we consider related work and summarize our main achievements.

2   The distributed π-calculus

In this section, we review Dπ, with some notational changes from [6]. And we do not consider type system since in this paper we are devoted to the encoding, thus the syntax, semantics and bisimulation relations are enough to achieve our aim.

2.1   syntax

Let $\mathcal{N}$ be a countable infinite set of names ranged over by the metavariable $x, y, z, \cdots$. We will distinguish two subsets of names: channels and locations, ranged over by $a, b, c, \cdots$ and $k, l, \cdots$ respectively. Let Net be the set of networks, ranged over by $M, N, \cdots$, built from processes, which in turn, are built from names. Fig. 1 lists the production grammar. All the operators for networks and processes have the usual meanings [2]. The definition of free names and bound names of a network N, written $fn(N)$ and $bn(N)$ respectively, arise as expected; the names of N, written $n(N)$, are given by $fn(N) \cup bn(N)$. Substitutions, ranged over by σ, are functions from $\mathcal{N}$ to $\mathcal{N}$; for any name e, $e\sigma$ represents the name by applying σ to e; we write $P\sigma$ and $N\sigma$ for the process and network obtained by applying σ to P and N respectively, with renaming possibly involved to avoid capture of free names. Note that Fig. 1 only displays monadic input and output, but we can easily extend them to allow a tuple of names $a_1, \cdots, a_n$ be transmitted. In some examples we write $\tilde{a}$ for the names occurring in polyadic input and output. The compound name introduced in [6] can also be added to our syntax.

To get an overview of the features of the language, we present an example borrowed from [2]. This example formalizes the mechanism of standard remote procedure call. A client process Q, located at k, wants to call a service located at l with arguments $\tilde{d}$, and waits for the results, to be processed by a continuation process P. Using our syntax, the example is written as $N = k[Q]|l[R]$, where

(12) $Q = (\nu r)(\text{go } l.\bar{a}\langle \tilde{d}, r, k\rangle | r(\tilde{v}).P)$
(13) $R = !a(\tilde{x}, y, z).\text{go } z.\bar{y}\langle f(\tilde{x})\rangle$

3

e	$::=$		names
		a	channel
	$\mid$	k	location
$P, Q, R \cdots$	$::=$		processes
		$\mathbf{0}$	inaction
	$\mid$	$P \mid Q$	composition
	$\mid$	$(\nu e)P$	restriction
	$\mid$	go $k.P$	movement
	$\mid$	$\bar{a}b.P$	output
	$\mid$	$a(x).P$	input
	$\mid$	$!P$	replication
	$\mid$	$[a = b]P, Q$	matching
$M, N \cdots$	$::=$		networks
		$\mathbf{0}$	empty
	$\mid$	$M \mid N$	composition
	$\mid$	$(\nu e)N$	restriction
	$\mid$	$k[P]$	agent

Fig. 1. Syntax of Dπ

2.2 Semantics

In [6] only reduction semantics is given. However we would like to use labelled transition semantics to compare the operational correspondence between Dπ and π_{rdi}. Fig. 2 displays this semantics in the so-called early style [8]. It is based on an auxiliary relation called structural congruence which abstract away some irrelevant details from the static structure of terms. It is closed under composition and restriction, satisfying the axioms listed below:

(i) $M \equiv N$ if M α-convertible N

(ii) $M \mid \mathbf{0} \equiv M$, $M \mid N \equiv N \mid M$, $M \mid (N \mid O) \equiv (M \mid N) \mid O$

(iii) $k[(\nu e)P] \equiv (\nu e)k[P]$ if $e \neq k$

(iv) $k[P \mid Q] \equiv k[P] \mid k[Q]$

(v) $(\nu e)\mathbf{0} \equiv \mathbf{0}$, $k[\mathbf{0}] \equiv \mathbf{0}$

4

$$\text{in } \frac{}{k[a(x).P] \xrightarrow{k,ab} k[P\{b/x\}]} \qquad \text{out } \frac{}{k[\bar{a}b.P] \xrightarrow{k,\bar{a}b} k[P]}$$

$$\text{go } \frac{}{k[\text{go } l.P] \xrightarrow{k,\tau} l[P]} \qquad \text{res } \frac{N \xrightarrow{k,\alpha} N' \quad b \notin n(\alpha) \cup \{k\}}{(\nu b)N \xrightarrow{k,\alpha} (\nu b)N'}$$

$$\text{open } \frac{N \xrightarrow{k,\bar{a}b} N' \quad b \notin \{k,a\}}{(\nu b)N \xrightarrow{k,\bar{a}(b)} N'} \qquad \text{mat } \frac{k[P] \xrightarrow{k,\alpha} k[P']}{k[[a=a]P,Q] \xrightarrow{k,\alpha} k[P']}$$

$$\text{mis } \frac{k[Q] \xrightarrow{k,\alpha} k[Q'] \quad a \neq b}{k[[a=b]P,Q] \xrightarrow{k,\alpha} k[Q']} \qquad \text{par } \frac{N \xrightarrow{k,\alpha} N' \quad bn(\alpha) \cap fn(M) = \emptyset}{N|M \xrightarrow{k,\alpha} N'|M}$$

$$\text{comm } \frac{M \xrightarrow{k,\bar{a}b} M' \quad N \xrightarrow{k,ab} N'}{M|N \xrightarrow{k,\tau} M'|N'} \qquad \text{close } \frac{M \xrightarrow{k,\bar{a}(b)} M' \quad N \xrightarrow{k,ab} N'}{M|N \xrightarrow{k,\tau} (\nu b)(M'|N')}$$

$$\text{rep } \frac{k[P|!P] \xrightarrow{k,\alpha} N}{k[!P] \xrightarrow{k,\alpha} N} \qquad \text{str } \frac{N \equiv M \quad M \xrightarrow{k,\alpha} M' \quad M' \equiv N'}{N \xrightarrow{k,\alpha} N'}$$

Fig. 2. Labelled transition system for Dπ

The rules com, close and par have a symmetric version which is omitted. Transitions are of the form $N \xrightarrow{k,\alpha} N'$, where k represents the location where the transition takes place and action α is the same as the action in standard π-calculus [10]. The definitions of $subj(\alpha)$, $fn(\alpha)$, $bn(\alpha)$ and $n(\alpha)$ inherit their meaning directly from the π-calculus. Relation $(\xrightarrow{k,\tau})^*$ is the reflexive and transitive closure of $\xrightarrow{k,\tau}$; moreover, $\xRightarrow{k,\alpha}$ stands for $(\xrightarrow{k,\tau})^* \xrightarrow{k,\alpha} (\xrightarrow{k,\tau})^*$, and $\xRightarrow{k,\hat{\alpha}}$ for $\xRightarrow{k,\alpha}$ if $\alpha \neq \tau$, and for $(\xrightarrow{k,\tau})^*$ if $\alpha = \tau$.

Remark. We explain why, in this transition system, a label (k, α) consists of two components, k and α. In rules comm and close, the prerequisite for the communication between M and N is that they can perform complementary actions in the *same* location. This kind of mechanism for communication is different from those of π_{1l}-calculus [1] and distributed join-calculus [5], where communication is deterministic, static and point-to-point, and participants for interaction need not be in the same location.

2.3 Bisimulation

We can now define a notion of (weak) bisimulation over the transition system given in last section. It is simply the easy adaption of the bisimulation relation in the π-calculus [8].

Definition 2.1 A symmetric relation $\mathcal{S} \subseteq Net \times Net$ is a bisimulation if

$M\mathcal{S}N$ implies:

If $M \xrightarrow{k,\alpha} M'$ then there exists N' such that $N \overset{k,\hat{\alpha}}{\Longrightarrow} N'$ and $M'\mathcal{S}N'$.

Two networks M and N are bisimilar, written $M \approx_D N$, if $M\mathcal{S}N$ for some bisimulation $\mathcal{S}$.

Another important notion in process calculi is expansion relation, since it is sometimes useful to count the number of silent moves performed by a process. Here we introduce this notion into Dπ and require that M expands to N if $M \approx_D N$and N has at least as many silent moves as M.

Definition 2.2 A relation $\mathcal{S} \subseteq Net \times Net$ is an expansion if $M\mathcal{S}N$ implies:
(1)If $M \xrightarrow{k,\alpha} M'$ then there exists some N' such that $N \overset{k,\alpha}{\Longrightarrow} N'$ and $M'\mathcal{S}N'$.
(2)If $N \xrightarrow{k,\alpha} N'$ then there exists some M' such that $M \xrightarrow{k,\hat{\alpha}} M'$ and $M'\mathcal{S}N'$.
N expands M, written $M \preceq_D N$, if $M\mathcal{S}N$ for some expansion $\mathcal{S}$.

Obviously, $M \preceq_D N$ implies $M \approx_D N$. It is easy to check that we also have the following lemma.

Lemma 2.3 $M \approx_D N$ *implies:*
(1) $M|O \approx_D N|O$ *(2)* $(\nu e)M \approx_D (\nu e)N$

3 π_{rdi}

In this section we introduce π_{rdi}, a restricted form of the π-calculus with delayed input. Since the processes in π_{rdi} is very similar to the processes located in some locations in Dπ, we abuse some notation and do not dwell much on the syntax and semantics of π_{rdi}.

Let Pr be the set of processes in π_{rdi}, also ranged over by $P, Q, \cdots$, built from the operators of *inaction, input prefix, output prefix, parallel composition, restriction, replication, match* and *delayed input* according to the following grammar:

$$P, Q, R \cdots ::= \mathbf{0} \mid a(x).P \mid \bar{a}b.P \mid P|Q \mid (\nu a)P \mid !P \mid [a = b]P, Q \mid a(x) : P$$

When the names transmitted are not important, we do not write them explicitly and abbreviate $a(x).P$, $\bar{a}b.P$ and $a(x) : P$ as $a.P$, $\bar{a}.P$ and $a : P$ respectively. The definitions of action α, $subj(\alpha)$, $fn(\alpha)$, $bn(\alpha)$, $n(\alpha)$ and $fn(P)$ remain unchanged. So the syntax of π_{rdi} is the same as that of π_{DI}.

The labelled transition semantics for π_{rdi} is displayed in Fig. 3, where u ranges over the set $\mathcal{N} \cup \{\varepsilon\}$. The semantics of π_{rdi} is obtained by applying three semantic constraints on π_{DI}. (1) $a(x) : P$ can receive a name along a only from one source: P, i.e., there is a subterm of P in the form of $\bar{a}b.Q$. The process $a(x) : P|\bar{a}b.R$ is not able to be reduced to $P\{b/x\}|R$ since $\bar{a}b.R$ is outside the scope of P. So we disallow the general rule for input:

$$\overline{a(x) : P \xrightarrow{ab} P\{b/x\}}$$

IN $\dfrac{}{a(x).P \xrightarrow{\varepsilon,ab} P\{b/x\}}$

OUT $\dfrac{}{\bar{a}b.P \xrightarrow{\varepsilon,\bar{a}b} P}$

ACT $\dfrac{P \xrightarrow{\varepsilon,\alpha} P' \quad x \notin n(\alpha) \quad a \notin subj(\alpha)}{a(x):P \xrightarrow{a,\alpha} a(x):P'}$

COM1 $\dfrac{P \xrightarrow{\varepsilon,\bar{a}b} P'}{a(x):P \xrightarrow{a,\tau} P'\{b/x\}}$

CLS1 $\dfrac{P \xrightarrow{\varepsilon,\bar{a}(b)} P'}{a(x):P \xrightarrow{a,\tau} (\nu b)(P'\{b/x\})}$

COM2 $\dfrac{P \xrightarrow{u,\bar{a}b} P' \quad Q \xrightarrow{u,ab} Q'}{P|Q \xrightarrow{u,\tau} P'|Q'}$

CLS2 $\dfrac{P \xrightarrow{u,\bar{a}(b)} P' \quad Q \xrightarrow{u,ab} Q' \quad b \notin fn(Q)}{P|Q \xrightarrow{u,\tau} (\nu b)(P'|Q')}$

PAR $\dfrac{P \xrightarrow{u,\alpha} P' \quad bn(\alpha) \cap fn(Q) = \emptyset}{P|Q \xrightarrow{u,\alpha} P'|Q}$

RES $\dfrac{P \xrightarrow{u,\alpha} P' \quad a \notin n(\alpha) \cup \{u\}}{(\nu a)P \xrightarrow{u,\alpha} (\nu a)P'}$

OPEN $\dfrac{P \xrightarrow{u,\bar{a}b} P' \quad b \notin \{u,a\}}{(\nu b)P \xrightarrow{u,\bar{a}(b)} P'}$

MAT $\dfrac{P \xrightarrow{u,\alpha} P'}{[a=b]P,Q \xrightarrow{u,\alpha} P'}$

MIS $\dfrac{Q \xrightarrow{u,\alpha} Q' \quad a \neq b}{[a=b]P,Q \xrightarrow{u,\alpha} Q'}$

REP $\dfrac{P|!P \xrightarrow{u,\alpha} P'}{!P \xrightarrow{u,\alpha} P'}$

STR $\dfrac{P \equiv Q \quad Q \xrightarrow{u,\alpha} Q' \quad Q' \equiv P'}{P \xrightarrow{u,\alpha} P'}$

Fig. 3. Labelled transition semantics for π_{rdi}

and only reserve COM1 and CLS1. (2) Suppose $P \equiv a(x):P_1$, $Q \equiv b(y):Q_1$ and P_1 has a complementary action of Q_1, then P can communicate with Q only if $a = b$. This reminds us of Remark 1. The encoding in Section 4 tells us the necessity of this constraint given by rules COM2 and CLS2. (3) We do not want two consecutively nesting delayed input to occur in a process expression. This constraint can be lifted, but keeping it simplifies our proofs in Section 4. Rule ACT implies this constraint.

The transition system rests on the following structural congruence relation.

(i) $P \equiv Q$ if P α-convertible Q

(ii) $P|\mathbf{0} \equiv P$, $P|Q \equiv Q|P$, $P|(Q|R) \equiv (P|Q)|R$

(iii) $(\nu a)\mathbf{0} \equiv \mathbf{0}$, $(\nu a)(\nu b)P \equiv (\nu b)(\nu a)P$

The label of a transition $P \xrightarrow{u,\alpha} P'$ has two components: u and α, where u is used for bookkeeping so as to reflect the second constraint. In fact, all the three constraints are imposed on the operator of delayed input. If we erase delayed input from π_{rdi}, u will always be ε. Moreover, in that case, when we write $P \xrightarrow{\varepsilon,\alpha} P'$ as $P \xrightarrow{\alpha} P'$, the LTS of π_{rdi} will become the LTS of standard π-calculus. So we can consider π as a special instance of π_{rdi}. Another point worthy of mentioning is that π_{rdi} is purely communication-based, involving neither one-dimensional mobility nor two-dimensional mobility.

Bisimulation and expansion can be adapted from Definitions 2.1 and 2.2 by replacing Dπ networks with π_{rdi} processes and extending the corresponding

$$
\begin{aligned}
\langle\!\langle \mathbf{0} \rangle\!\rangle &= \mathbf{0} \\
\langle\!\langle M|N \rangle\!\rangle &= \langle\!\langle M \rangle\!\rangle|\langle\!\langle N \rangle\!\rangle \\
\langle\!\langle (\nu e)N \rangle\!\rangle &= (\nu e)\langle\!\langle N \rangle\!\rangle \\
\langle\!\langle k[P] \rangle\!\rangle &= \langle\!\langle P \rangle\!\rangle^k
\end{aligned}
\qquad
\begin{aligned}
\langle\!\langle \mathbf{0} \rangle\!\rangle^k &= \mathbf{0} \\
\langle\!\langle P|Q \rangle\!\rangle^k &= \langle\!\langle P \rangle\!\rangle^k|\langle\!\langle Q \rangle\!\rangle^k \\
\langle\!\langle (\nu e)P \rangle\!\rangle^k &= (\nu e)\langle\!\langle P \rangle\!\rangle^k \\
\langle\!\langle \mathsf{go}\ l.P \rangle\!\rangle^k &= k:\bar{k}.\langle\!\langle P \rangle\!\rangle^l \\
\langle\!\langle \bar{a}b.P \rangle\!\rangle^k &= k:\bar{a}b.\bar{k}.\langle\!\langle P \rangle\!\rangle^k \\
\langle\!\langle a(x).P \rangle\!\rangle^k &= k:a(x).\bar{k}.\langle\!\langle P \rangle\!\rangle^k \\
\langle\!\langle !P \rangle\!\rangle^k &= !\langle\!\langle P \rangle\!\rangle^k \\
\langle\!\langle [a=b]P,Q \rangle\!\rangle^k &= [a=b]\langle\!\langle P \rangle\!\rangle^k, \langle\!\langle Q \rangle\!\rangle^k
\end{aligned}
$$

Fig. 4. Encoding Dπ into π_{rdi}

definition of $\overset{u,\hat{a}}{\Longrightarrow}$.

Definition 3.1 A symmetric relation $\mathcal{S} \subseteq Pr \times Pr$ is a bisimulation if $P\mathcal{S}Q$ implies:
If $P \overset{u,\alpha}{\longrightarrow} P'$ then there exists Q' such that $Q \overset{u,\hat{\alpha}}{\Longrightarrow} Q'$ and $P'\mathcal{S}Q'$
P is bisimilar to Q, written $P \approx_r Q$, if $P\mathcal{S}Q$ for some $\mathcal{S}$.

Definition 3.2 A relation $\mathcal{S} \subseteq Pr \times Pr$ is an expansion if $P\mathcal{S}Q$ implies:
(1) If $P \overset{u,\alpha}{\longrightarrow} P'$ then there exists some Q' such that $Q \overset{u,\alpha}{\Longrightarrow} Q'$ and $P'\mathcal{S}Q'$.
(2) If $Q \overset{u,\alpha}{\longrightarrow} Q'$ then there exists some P' such that $P \overset{u,\hat{\alpha}}{\longrightarrow} P'$ and $P'\mathcal{S}Q'$.
Q expands P, written $P \preceq_r Q$, if $P\mathcal{S}Q$ for some expansion $\mathcal{S}$.

All the properties of $\approx_D$ and $\preceq_D$ hold for $\approx_r$ and $\preceq_r$.

4   Encoding Dπ into π_{rdi}

The translation from Dπ to π_{rdi} is displayed in Fig. 4, where the translation of networks relies on an auxiliary translation of processes which is parametric in a location name. The name represents the location where the process is running. Every location is translated into a delayed input and an output within its continuation process, and we are not concerned with what is transmitted between them. The pair of delayed input and the continuing output performs as an entity, either appear or disappear at the same time. That's why we impose constraints (1) and (2) on π_{DI} so as to reduce it to π_{rdi}.

Let's consider the network N defined in Section 2.1. It can be translated as $\langle\!\langle N \rangle\!\rangle = \langle\!\langle Q \rangle\!\rangle^k|\langle\!\langle R \rangle\!\rangle^l$, where

(14) $\langle\!\langle Q \rangle\!\rangle^k = (\nu r)(k:\bar{k}.l:\bar{a}\langle\tilde{d},r,k\rangle.\bar{l}|k:r(\tilde{v}).\bar{k}.\langle\!\langle P \rangle\!\rangle^k)$

8

(15) $\langle\!\langle R \rangle\!\rangle^l = !l : a(\tilde{x}, y, z).\bar{l}.l : \bar{l}.z : \bar{y}\langle f(\tilde{x})\rangle.\bar{z}$

For the translation $\langle\!\langle N \rangle\!\rangle$, we neither add any new names to nor delete any existing names from N, so the free names of N are preserved and remain free in $\langle\!\langle N \rangle\!\rangle$.

Lemma 4.1 $fn(\langle\!\langle M \rangle\!\rangle) = fn(M)$

Lemma 4.2 *For any substitution σ, the following properties hold:*
(1) $(\langle\!\langle P \rangle\!\rangle^k)\sigma = \langle\!\langle P\sigma \rangle\!\rangle^{k\sigma}$
(2) $\langle\!\langle M\sigma \rangle\!\rangle = \langle\!\langle M \rangle\!\rangle\sigma$

Proof. By induction on the structures of processes and networks. The proof of clause (2) makes use of (1) when analyzing the case of $M \equiv k[P]$. $\square$

Lemma 4.3 *(1) If $M \equiv N$ then $\langle\!\langle M \rangle\!\rangle \equiv \langle\!\langle N \rangle\!\rangle$.*
(2) If $\langle\!\langle M \rangle\!\rangle \equiv P$ then there exists some N such that $P \equiv \langle\!\langle N \rangle\!\rangle$ and $M \equiv N$.

Proof. By induction on the structural congruence of Net and Pr respectively. Details are omitted. $\square$

Now we are prepared to prove that the encoding $\langle\!\langle \cdot \rangle\!\rangle$ enjoys an operational correspondence up to $\preceq_r$.

Lemma 4.4 (Operational correspondence) *Let M be a network in $D\pi$.*
(1) Suppose that $M \xrightarrow{k,\alpha} M'$, then we have:
(a) if $\alpha \in \{ab, \bar{a}b, \bar{a}(b)\}$ then $\langle\!\langle M \rangle\!\rangle \xrightarrow{k,\alpha}\xrightarrow{k,\tau} \langle\!\langle M' \rangle\!\rangle$;
(b) if $\alpha = \tau$ then $\langle\!\langle M \rangle\!\rangle(\xrightarrow{k,\tau})^t \langle\!\langle M' \rangle\!\rangle$ and $t = 1$ or $t = 3$.
(2) Suppose that $\langle\!\langle M \rangle\!\rangle \xrightarrow{u,\alpha} P$, then there exists $M' \in Net$ such that $M \xrightarrow{u,\alpha} M'$ and $\langle\!\langle M' \rangle\!\rangle \preceq_r P$.

Proof. By transition induction. The proof relies on the last three lemmas. Details can be found in the Appendix. $\square$

Theorem 4.5 (Full abstraction) *Let M and N be two networks in $D\pi$, then*

$$M \approx_D N \text{ iff } \langle\!\langle M \rangle\!\rangle \approx_r \langle\!\langle N \rangle\!\rangle$$

Proof. ($\Rightarrow$) We can construct a relation

$$\mathcal{R} = \{(\langle\!\langle M \rangle\!\rangle, \langle\!\langle N \rangle\!\rangle) \mid M \approx_D N\} \cup \approx_r$$

and prove that it is a bisimulation in π_{rdi}.
($\Leftarrow$) Similarly we can prove that the relation

$$\mathcal{S} = \{(M, N) \mid \langle\!\langle M \rangle\!\rangle \approx_r \langle\!\langle N \rangle\!\rangle\}$$

is a bisimulation in $D\pi$.

The two parts of proof rely on Lemma 4.4. Refer to the Appendix for more details. $\square$

5 Discussion

We take mobile ambients [4] as a typical example of process calculus with two-dimensional mobility. It has two characteristics: (1) it has hierarchical structure, i.e., an ambient may contain several subambients; (2) a process may be influenced by the operations of its sibling processes without noticing the effect. For example, the following reduction rules are the cornerstones for ambient calculus.

(16) $n[\text{in } m.P|Q]|m[R] \longrightarrow m[n[P|Q]|R]$

(17) $m[n[\text{out } m.P|Q]|R] \longrightarrow n[P|Q]|m[R]$

In (16) and (17), the environments of Q are both changed by P. There is not any symptom for Q to know when and how its environment will be changed. This kind of phenomenon is hard to be simulated in communication-based models, since in these models a process cannot have any influence to a sibling process if no communication happens between them. So one may think of treating a pair of sibling processes, say $\text{in } m.P$ and Q in (16) as a whole entity in the translation (if it exists), and either process can perform some actions to influence the entity, which also means to influence itself and its sibling process. There are two reasons against this strategy. (1) A good translation should be homomorphic with respect to parallel composition, namely $\langle\!\langle P|Q\rangle\!\rangle = \langle\!\langle P\rangle\!\rangle|\langle\!\langle Q\rangle\!\rangle$ [9]. This condition requires us to encode $\text{in } m.P$ and Q separately. (2) In (17), the behavior of ambient n has no effect to the sibling process R. Therefore, it should be better to take ambient n and R as two different entities in the translation. Now we are trapped in a dilemma: there are arguments to support two opposite opinions.

To translate two-dimensional mobility, we have to consider the problem of how to represent hierarchical structure in a flat space. (In the π-calculus and its variants, all processes stand on the same level.) It is difficult to use only prefix operators to describe nesting ambients. The techniques of polyadic synchronization [3] can be thought of communication over channels with structured address. But it is of little help for the translation of ambient. For example, one may conceive of a translation of m in (17) like this: $m \cdot n : C[\langle\!\langle\text{out } m.P|Q\rangle\!\rangle]|m : D[\langle\!\langle R\rangle\!\rangle]$, where C and D are some contexts of the target language. It says that C and D lie on different levels. But it does not solve the problem. If R has the operator of $\text{open } n$, one may want C to be lifted to a higher level, directly underneath m. However, the semantics of polyadic synchronization [3] forbids any communication between two processes sharing different channels, say $m \cdot n$ and n in this example.

Therefore, we leave the proof or disproof of the possibility of directly encoding two-dimensional mobility into communication-based models as an open problem.

6 Conclusions and related work

We have presented a variant of the π-calculus (π_{rdi}) by imposing three semantic constraints on the π-calculus with delayed input. The model we obtained is purely based on communication, involving nothing about mobility. By encoding Hennessy and Riely's distributed π-calculus ($D\pi$) into π_{rdi}, we show the possibility of characterizing one-dimensional mobility using only communication. To compare the operational correspondence between $D\pi$ and π_{rdi}, we enrich the original $D\pi$ with labelled transition semantics and bisimulation. It is shown that the encoding is fully abstract. We have also discussed the difficulties of dealing with two-dimensional mobility which is one of the main features of mobile ambients.

In [1], Amadio has presented an asynchronous model (π_{1l}) of locality, failure and process mobility. It is shown that π_{1l} can be encoded into π_1, a typed version of the asynchronous π-calculus. But π_{1l} requires that every channel name is associated with a unique process which serves messages addressed to that name. Therefore, location does not have a strong efficacy to limit resource access, since the communication is point-to-point and the processes of sending and receiving message can stay in different locations. In contrast, the communication in $D\pi$ is local and there is no direct message exchange between two different locations. In fact, the main role location plays in π_{1l} is to describe failure. In the absence of failure, the distribution of processes is transparent and enjoys the property: $r \approx_a er_l(r)$, where er_l is the function of location erasure [1]. In [5], location nesting shows the hierarchical structure of location tree. But message transmission is also deterministic and point-to-point as every name is defined in at most one solution. Hence it is not surprising that there exists an encoding of distributed join-calculus into the join-calculus. In [2], a subcalculus of $D\pi$, the receptive distributed π-calculus is studied. It is shown to be expressive enough to contain the π_1-calculus proposed in [1].

References

[1] R. M. Amadio. An asynchronous model of locality, failure, and process mobility. In D. Garlan and D. Le Metayer, editors, *Proceedings of COORDINATION '97*, volume 1282 of *LNCS*. Springer, 1997. Extended version as Rapport de Recherche RR-3109, INRIA Sophia-Antipolis, 1997.

[2] R. M. Amadio, G. Boudol, and C. Lhoussaine. The receptive distributed pi-calculus (extended abstract). In C. Pandu Rangan, V. Raman, and R. Ramanujam, editors, *Proceedings of FSTTCS '99*, volume 1738 of *LNCS*, pages 304–315. Springer, Dec. 1999.

[3] M. Carbone. On the expressive power of polyadic synchronisation in π-calculus. *Electronic Notes in Theoretical Computer Science*, 68(2), 2002.

[4] L. Cardelli and A. D. Gordon. Mobile ambients. *Theoretical Computer Science*, 240(1):177–213, 2000. An extended abstract appeared in *Proceedings of FoSSaCS '98*: 140–155.

[5] C. Fournet, G. Gonthier, J.-J. Lévy, L. Maranget, and D. Rémy. A calculus of mobile agents. In U. Montanari and V. Sassone, editors, *Proceedings of CONCUR '96*, volume 1119 of *LNCS*, pages 406–421. Springer, 1996.

[6] M. Hennessy and J. Riely. Resource access control in systems of mobile agents. In U. Nestmann and B. C. Pierce, editors, *Proceedings of HLCL '98*, volume 16.3 of *ENTCS*, pages 3–17. Elsevier Science Publishers, 1998. Full version as CogSci Report 2/98, University of Sussex, Brighton.

[7] M. Merro. *Locality in the π-calculus and applications to distributed objects.* PhD thesis, Ecole des Mines, France, October 2000.

[8] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, part I/II. *Journal of Information and Computation*, 100:1–77, Sept. 1992.

[9] C. Palamidessi. Comparing the expressive power of the synchronous and the asynchronous π-calculus. In *Proceedings of POPL '97*, pages 256–265. ACM, Jan. 1997.

[10] D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes.* Cambridge University Press, 2001.

A Delayed Proofs

Proof of Lemma 4.4

Proof. The proof proceeds by transition induction. We prove part 1. The proof of part 2 is similar.

(i) Suppose that $M \xrightarrow{k,\alpha} M'$ and $\alpha = ab$. When $\alpha = \bar{a}b$ or $\bar{a}(b)$, similar analysis can be applied.

    (a) When the last rule applied to get $M \xrightarrow{k,ab} M'$ is in, then we have $\langle\!\langle M \rangle\!\rangle \equiv k : a(x).\bar{k}.\langle\!\langle P \rangle\!\rangle^k \xrightarrow{k,ab} k : \bar{k}.\langle\!\langle P \rangle\!\rangle^k \{b/x\} \xrightarrow{k,\tau} \langle\!\langle P \rangle\!\rangle^k \{b/x\}$. By Lemma 4.2, $\langle\!\langle P \rangle\!\rangle^k \{b/x\} = \langle\!\langle P\{b/x\} \rangle\!\rangle^k \equiv \langle\!\langle M' \rangle\!\rangle$.

    (b) When res is the rule used last, then $M \equiv (\nu e)N, M' \equiv (\nu e)N'$, $e \notin \{a, b\}$ and $N \xrightarrow{k,ab} N'$. By induction hypothesis, $\langle\!\langle N \rangle\!\rangle \xrightarrow{k,ab} Q \xrightarrow{k,\tau} \langle\!\langle N' \rangle\!\rangle$, it follows that $(\nu e)\langle\!\langle N \rangle\!\rangle \xrightarrow{k,ab} (\nu e)Q \xrightarrow{k,\tau} (\nu e)\langle\!\langle N' \rangle\!\rangle \equiv \langle\!\langle M' \rangle\!\rangle$.

    (c) When the last rule used is mat, then $M \equiv k[[c = c]P, Q]$, $M' \equiv k[P']$ and $k[P] \xrightarrow{k,ab} k[P']$ for some c, P and Q. By induction hypothesis, we can obtain $\langle\!\langle P \rangle\!\rangle^k \xrightarrow{k,ab} R \xrightarrow{k,\tau} \langle\!\langle P' \rangle\!\rangle^k$, then it follows that $\langle\!\langle M \rangle\!\rangle \equiv [c = c]\langle\!\langle P \rangle\!\rangle^k, \langle\!\langle Q \rangle\!\rangle^k \xrightarrow{k,ab} R \xrightarrow{k,\tau} \langle\!\langle P' \rangle\!\rangle^k \equiv \langle\!\langle M' \rangle\!\rangle$ by rule MAT.

    (d) The case for mis is similar to the last case.

(e) When par is used last to derive the transition of M, then $M \equiv N_1 | N_2$, $M' \equiv N_1' | N_2$ and $N_1 \xrightarrow{k,ab} N_1'$, for some N_1 and N_2. By induction hypothesis, $\langle\!\langle N_1 \rangle\!\rangle \xrightarrow{k,ab} R \xrightarrow{k,\tau} \langle\!\langle N_1' \rangle\!\rangle$. It follows that $\langle\!\langle M \rangle\!\rangle \equiv \langle\!\langle N_1 \rangle\!\rangle | \langle\!\langle N_2 \rangle\!\rangle \xrightarrow{k,ab} R | \langle\!\langle N_2 \rangle\!\rangle \xrightarrow{k,\tau} \langle\!\langle N_1' \rangle\!\rangle | \langle\!\langle N_2 \rangle\!\rangle \equiv \langle\!\langle M' \rangle\!\rangle$ by rule PAR.

(f) When the last rule applied is rep, then $M \equiv k[!P]$ and $k[P|!P] \xrightarrow{k,ab} M'$. By induction hypothesis, $\langle\!\langle P \rangle\!\rangle^k | ! \langle\!\langle P \rangle\!\rangle^k \xrightarrow{k,ab} R \xrightarrow{k,\tau} \langle\!\langle M' \rangle\!\rangle$. By rule REP, we have that $\langle\!\langle M \rangle\!\rangle \equiv \langle\!\langle P \rangle\!\rangle^k \xrightarrow{k,ab} R \xrightarrow{k,\tau} \langle\!\langle M' \rangle\!\rangle$.

(g) When str is applied as the last rule, then $M \equiv N$, $N \xrightarrow{k,ab} N'$ and $N' \equiv M'$. From induction hypothesis, we have $\langle\!\langle N \rangle\!\rangle \xrightarrow{k,ab} \xrightarrow{k,\tau} \langle\!\langle N' \rangle\!\rangle$. Lemma 4.3 shows that $\langle\!\langle M \rangle\!\rangle \equiv \langle\!\langle N \rangle\!\rangle$, $\langle\!\langle N' \rangle\!\rangle \equiv \langle\!\langle M' \rangle\!\rangle$. By using rule STR, we conclude that $\langle\!\langle M \rangle\!\rangle \xrightarrow{k,ab} \xrightarrow{k,\tau} \langle\!\langle M' \rangle\!\rangle$.

(ii) $\alpha = \tau$

(a) When the rule used to derive the transition of M is go, then $M \equiv k[\text{go } l.P]$, $N \equiv l[P]$ for some l and P. So we have $\langle\!\langle M \rangle\!\rangle = k : \bar{k}.\langle\!\langle P \rangle\!\rangle^l \xrightarrow{k,\tau} \langle\!\langle P \rangle\!\rangle^l \equiv \langle\!\langle N \rangle\!\rangle$.

(b) When com is applied last, then $M \equiv N_1 | N_2$, $M' \equiv N_1' | N_2'$, $N_1 \xrightarrow{k,\bar{a}b} N_1'$ and $N_2 \xrightarrow{k,ab} N_2'$ for some N_1, N_1', N_2 and N_2'. By induction hypothesis, $\langle\!\langle N_1 \rangle\!\rangle \xrightarrow{k,\bar{a}b} R_1 \xrightarrow{k,\tau} \langle\!\langle N_1' \rangle\!\rangle$, $\langle\!\langle N_2 \rangle\!\rangle \xrightarrow{k,ab} R_2 \xrightarrow{k,\tau} \langle\!\langle N_2' \rangle\!\rangle$. So $\langle\!\langle M \rangle\!\rangle \equiv \langle\!\langle N_1 \rangle\!\rangle | \langle\!\langle N_2 \rangle\!\rangle (\xrightarrow{k,\tau})^3 \langle\!\langle N_1' \rangle\!\rangle | \langle\!\langle N_2' \rangle\!\rangle \equiv \langle\!\langle M_2' \rangle\!\rangle$ by rule COM2, PAR.

(c) When the rule close is applied in the last step of derivation, then $M \equiv N_1 | N_2$, $M' \equiv (\nu b)(N_1' | N_2')$, $N_1 \xrightarrow{k,\bar{a}(b)} N_1'$ and $N_2 \xrightarrow{k,ab} N_2'$, for some N_1, N_1', N_2 and N_2'. By induction hypothesis $\langle\!\langle N_1 \rangle\!\rangle \xrightarrow{k,\bar{a}(b)} R_1 \xrightarrow{k,\tau} \langle\!\langle N_1' \rangle\!\rangle$ and $\langle\!\langle N_2 \rangle\!\rangle \xrightarrow{k,ab} R_2 \xrightarrow{k,\tau} \langle\!\langle N_2' \rangle\!\rangle$. So we have $\langle\!\langle M \rangle\!\rangle \equiv \langle\!\langle N_1 \rangle\!\rangle | \langle\!\langle N_2 \rangle\!\rangle (\xrightarrow{k,\tau})^3 (\nu b)(\langle\!\langle N_1' \rangle\!\rangle | \langle\!\langle N_2' \rangle\!\rangle) \equiv \langle\!\langle M' \rangle\!\rangle$ by rule CLS2, PAR and RES.

(d) The other cases are similar to 1.

$\square$

Proof of Theorem 4.5

Proof. ($\Rightarrow$) We can construct a relation

$$\mathcal{R} = \{(\langle\!\langle M \rangle\!\rangle, \langle\!\langle N \rangle\!\rangle) \mid M \approx_D N\}$$

and prove that it is a bisimulation up to $\approx_r$ in π_{rdi}. Suppose that $\langle\!\langle M \rangle\!\rangle \xrightarrow{k,\alpha} P$, $\alpha \in \{ab, \bar{a}b, \bar{a}(b), \tau\}$. By operational correspondence, we have $M \xrightarrow{k,\alpha} M'$ for some M' such that $\langle\!\langle M' \rangle\!\rangle \preceq_r P$, i.e., $\langle\!\langle M' \rangle\!\rangle \approx_r P$. Since $M \approx_D N$, we know that $N \xRightarrow{k,\hat{\alpha}} N'$ and $M' \approx_D N'$. By Lemma 4.4, $\langle\!\langle N \rangle\!\rangle \xRightarrow{k,\hat{\alpha}} \langle\!\langle N' \rangle\!\rangle \equiv Q$. So we have $P \approx_r \mathcal{R} Q$. Similarly, by using the "up-to" techniques of [10], we can prove that the relation $\mathcal{R} \cup \approx_r$ is a bisimulation in π_{rdi}.

13

($\Longleftarrow$) Similarly we can prove that the relation

$$\mathcal{S} = \{(M, N) \mid \langle\!\langle M \rangle\!\rangle \approx_r \langle\!\langle N \rangle\!\rangle\}$$

is a bisimulation in Dπ. Suppose $M \xrightarrow{k,\alpha} M'$. By operational correspondence we have $\langle\!\langle M \rangle\!\rangle \xRightarrow{k,\alpha} \langle\!\langle M' \rangle\!\rangle$. Since $\langle\!\langle M \rangle\!\rangle \approx_r \langle\!\langle N \rangle\!\rangle$, we know that $\langle\!\langle N \rangle\!\rangle \xRightarrow{k,\hat{\alpha}} P \approx_r \langle\!\langle M' \rangle\!\rangle$. From Lemma 4.4, it follows that there exists some N', such that $N \xRightarrow{k,\hat{\alpha}} N'$ and $\langle\!\langle N' \rangle\!\rangle \preceq_r P$. So we have that $\langle\!\langle N' \rangle\!\rangle \approx_r P \approx_r \langle\!\langle M' \rangle\!\rangle$. Therefore $(M',\ N') \in \mathcal{S}$, which means that $\mathcal{S}$ is a bisimulation in Dπ.

<div align="right">□</div>