The Seal Calculus Revisited: contextual equivalence and bisimilarity *

Giuseppe Castagna and Francesco Zappa Nardelli

LIENS (CNRS), 45, rue d'Ulm, 75005 Paris, France.
{castagna,zappa}@ens.fr

Abstract. We present a new version of the Seal Calculus, a calculus of mobile computation. We study observational congruence and bisimulation theory, and show how they are related.

1 Introduction

The Seal Calculus is a calculus of mobile computations conceived to model secure programming of large scale distributed systems over open networks. It can be roughly described as the π-calculus [9] with hierarchical location mobility and remote accesses to resources. The original presentation [14] was tailored with an implementation in mind, and offered features that were important from the practical view point (e.g. portals), but unessential for its theoretical study.

In this paper we present a 'revised' version of the Seal Calculus that shares with the original version part of its syntax and all the design guiding principles, while gets rid of the redundant aspects. In particular, portals do not belong to the calculus anymore, the reduction semantics no longer resorts to an auxiliary relation, new security oriented rules handle the extrusion of private names, and the definition of the calculus is parametric on the semantics of remote interaction, thus allowing an easier exploration of the design space.

We concentrate on behavioural equivalences to establish more than a foundation to it, by proving that a bisimulation-based equivalence is sound with respect to weak barbed congruence. Even if the technique used is standard, this work is important because it is the first one that keeps into account agent duplication:[1] avoiding agent duplication hugely simplifies the study of the calculus, but wipes out many properties that characterise real systems.

2 Syntax and Semantics of Seal

The syntax of Seal is reported in Figure 2, where letters u, v, x, y, z range over variables, P, Q, R, S range over processes, and $n \geq 0$. The syntax of processes is the same of Ambient Calculus [2]: the only remark is that replication is guarded. On the other hand, contrary to what happens in Ambients, all interactions take

* Research supported by 'MyThS: Models and Types for Security in Mobile Distributed Systems', EU FET-GC IST-2001-32617.

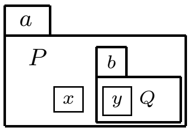
[1] Sangiorgi's research on equivalences for higher order π-calculus [10,11] allows the duplication of processes, but does not account either for agents or for mobility.

Processes			**Actions**			**Locations**		
P ::=	$\mathbf{0}$	inactivity	α ::=	$\overline{x}^{\eta}(y_1, \cdots, y_n)$	output	η ::=	$*$	local
$\mid$	$P \mid P$	composition	$\mid$	$x^{\eta}(y_1, \cdots, y_n)$	input	$\mid$	$\uparrow$	up
$\mid$	$!\,\gamma.P$	replication	$\mid$	$\overline{x}^{\eta}\{y\}$	send	$\mid$	z	down
$\mid$	$(\boldsymbol{\nu}\, x)\, P$	restriction	$\mid$	$x^{\eta}\{y_1, \cdots, y_n\}$	receive	**Guards**		
$\mid$	$\alpha.P$	action				γ ::=	$\overline{x}^{*}()$	
$\mid$	$x[\,P\,]$	seal				$\mid$	$x^{*}()$	

Fig. 1. Syntax of the Seal Calculus

place on named localised channels.

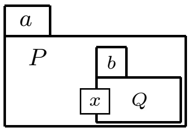
In this work we present two different dialects of Seal. In the first one, called *Located Seal*, channels are *located* inside seals.



Located channels

Channel denotations specify in which seal a channel is located: a channel named x is denoted by x^{η}, where η is $*$ when the channel is local, is $\uparrow$ when the channel is in the parent, and is n when the channel is in a child seal n. The figure on the left represents a located channels situation with two channels x and y, the former located in a and the latter in b. A synchronisation between P and Q can happen on any of these channels. In order to synchronise on x, process P will use x^* as for P channel x is local, while Q will use $x^{\uparrow}$ as it is a channel located in the parent. Similarly, to synchronise on y, P will use y^b and Q will use y^*.

In the second dialect, called *Shared Seal*, channels are *shared* between the two communicating agents in parent-child relation, so that the η represents the partner the channel is shared with. Thus, $x^{\uparrow}$ denotes the channel x shared with the parent seal, x^n the denotes the channel x shared with the child n, while x^* still denotes a local channel. The figure on the left represents a shared channel case. To synchronise, P and Q must use a channel shared between a and b, such as x. For such a synchronisation, P will use x^b as it is a channel shared with b, and Q will use $x^{\uparrow}$ as it is a channel shared with the parent.



Shared channels

In order to give reduction rules that are parametric on the interaction pattern being used, we introduce two predicates $\mathsf{synch}^S, \mathsf{synch}^L$: $\mathbf{Var} \times \mathbf{Loc} \times \mathbf{Loc} \rightarrow \mathbf{Bool}$, ranged over by synch. Intuitively, $\mathsf{synch}_y(\eta_1, \eta_2)$ holds if and only if for any channel x an action on x^{η_1} performed in some parent seal may synchronise with a coaction on x^{η_2} performed in a child seal y.

Definition 1. *Let η_1, η_2 be locations and y a variable (a seal name). We define:*
1. $\mathsf{synch}^S_y(\eta_1, \eta_2) \stackrel{def}{=} (\eta_1 = y \wedge \eta_2 = \uparrow)$ [*Shared* Seal]
2. $\mathsf{synch}^L_y(\eta_1, \eta_2) \stackrel{def}{=} (\eta_1 = y \wedge \eta_2 = *) \vee (\eta_1 = * \wedge \eta_2 = \uparrow)$ [*Located* Seal]

Channel synchronisation is used for the two possible forms of interaction:

*Communication***:** $\overline{x}^{\eta}(\vec{y}).P$ denotes a process waiting to output $\vec{y}$ on channel x^{η} and then behave like P; $x^{\eta}(\vec{y}).P$ denotes a process waiting to read on channel

x^η some input, say $\vec{z}$, and then behave like $P\{\vec{z}/\vec{y}\}$, that is P in which z_i is substituted for every free occurrence of y_i;

Mobility: $\overline{x}^\eta\langle\!\{y\}\!\rangle.P$ denotes a process waiting to serialise a child seal named y, send it along channel x^η and then behave like P; $x^\eta\langle\!\{\vec{z}\}\!\rangle.P$ denotes a process waiting to receive one seal body along channel x^η, to reactivate n identical copies of it under the names $z_1, \ldots, z_n$ and then behave like P.

The semantics of the Seal Calculus is given in terms of a structural congruence relation and a set of reduction rules. We write $\vec{x}_n$ or just $\vec{z}$ to denote the tuple $x_1, \cdots, x_n$, $(\boldsymbol{\nu}\, \vec{x}_n)$, or just $(\boldsymbol{\nu}\, \vec{x})$, as an abbreviation for $(\boldsymbol{\nu}\, x_1) \ldots (\boldsymbol{\nu}\, x_n)$, and omit trailing $\mathbf{0}$ processes. We work modulo α-conversion, and require the y_i to be pairwise distinct in the input action. The definition of the set of free variables of a process is standard, except for the receive action that is not a binding operation ($fv(x^\eta\langle\!\{\vec{y}\}\!\rangle.P) = fv(P) \cup \vec{y} \cup \{x\} \cup fv(\eta)$), and coincides with the one in [14].

Definition 2 (Structural Congruence). *The structural congruence relation $\equiv$ is the smallest congruence over processes that makes $(P/\equiv, \mid, \mathbf{0})$ a commutative monoid and satisfies the following axioms: (1) $(\boldsymbol{\nu}\, x)\, \mathbf{0} \equiv \mathbf{0}$; (2) $(\boldsymbol{\nu}\, x)\, (\boldsymbol{\nu}\, y)\, P \equiv (\boldsymbol{\nu}\, y)\, (\boldsymbol{\nu}\, x)\, P$ for $x \neq y$; (3) $(\boldsymbol{\nu}\, x)\, (P \mid Q) \equiv P \mid (\boldsymbol{\nu}\, x)\, Q$ for $x \notin fv(P)$; (4) $!P \equiv P \mid\, !P$.*

In the process $(\boldsymbol{\nu}\, \vec{x})\, P$, we can suppose $x_1, \ldots, x_n$ to be pairwise distinct, and freely permute them (axiom 2 of Definition 2). This implies that the vector $\vec{x}$ behaves as a set, thus justifying notations such as $(\boldsymbol{\nu}\, \vec{x} \cap \vec{y})\, P$ or $(\boldsymbol{\nu}\, \vec{x} \setminus \vec{y})\, P$ (where $\cap$ and $\setminus$ denote set-theoretic intersection and difference, with the convention that $(\boldsymbol{\nu}\, \varnothing)\, P = P$).

Definition 2 is the standard π-calculus structural congruence definition. It should be remarked that the Ambient's axiom:

$$(\boldsymbol{\nu}\, x)\, y[\, P\,] \equiv y[\, (\boldsymbol{\nu}\, x)\, P\,] \qquad \text{for } x \neq y \qquad (*)$$

is not (and must not be) used in Seal. This is due to the presence, in Seal, of duplication: it would be semantically unsound to define the processes $(\boldsymbol{\nu}\, x)\, y[\, P\,]$ and $y[\, (\boldsymbol{\nu}\, x)\, P\,]$ as equivalent in the presence of duplication, since if we compose both terms with the copier process $Q = \mathsf{copy}\ y\ \mathsf{as}\ z$ (whose definition can be found in the next page) we obtain: $y[\, (\boldsymbol{\nu}\, x) P\,] \mid Q \rightarrow y[\, (\boldsymbol{\nu}\, x) P\,] \mid z[\, (\boldsymbol{\nu}\, x) P\,]$ and $(\boldsymbol{\nu}\, x)\, y[\, P\,] \mid Q \rightarrow (\boldsymbol{\nu}\, x)\, (y[\, P\,] \mid z[P])$ The first process yields a configuration where seals y and z have each a private channel x, while the second process produces a configuration where y and z share a common channel x.

This observation holds true independently from the Seal framework: the extrusion rule $(*)$ is authorised in Ambient only because its definition does not allow ambients duplication. Among its consequences, it is worth stressing that the extrusion of locally restricted names, when allowed, must be handled explicitly by the reduction rules. The approach we choose is to extrude all locally restricted variables that are communicated to the parent, and no other. This is obtained by the reduction rules shown in Figure 2 to which the usual rules for context and structural congruence reduction must also be added.

$$x^*(\vec{u}).P \mid \overline{x}^*(\vec{v}).Q \rightarrow P\{^{\vec{v}}/_{\vec{u}}\} \mid Q$$

$$\overline{x}^{\eta_1}(\vec{v}).P \mid y[\,(\boldsymbol{\nu}\,\vec{z})\,(x^{\eta_2}(\vec{u}).Q_1 \mid Q_2)\,] \rightarrow P \mid y[\,(\boldsymbol{\nu}\,\vec{z})\,(Q_1\{^{\vec{v}}/_{\vec{u}}\} \mid Q_2)\,] \quad \text{if } \vec{v}\cap\vec{z}=\varnothing$$

$$x^{\eta_1}(\vec{u}).P \mid y[\,(\boldsymbol{\nu}\,\vec{z})\,(\overline{x}^{\eta_2}(\vec{v}).Q_1 \mid Q_2)\,] \rightarrow (\boldsymbol{\nu}\,\vec{v}\cap\vec{z})\,(P\{^{\vec{v}}/_{\vec{u}}\} \mid y[\,(\boldsymbol{\nu}\,\vec{z}\setminus\vec{v})\,(Q_1 \mid Q_2)\,])$$

$$x^*\{\vec{u}\}.P_1 \mid \overline{x}^*\{v\}.P_2 \mid v[\,Q\,] \rightarrow P_1 \mid u_1[\,Q\,] \mid \cdots \mid u_n[\,Q\,] \mid P_2$$

$$\overline{x}^{\eta_1}\{v\}.P \mid v[R] \mid y[(\boldsymbol{\nu}\,\vec{z})(x^{\eta_2}\{\vec{u}\}.Q_1 \mid Q_2)\,] \rightarrow P \mid y[\,(\boldsymbol{\nu}\,\vec{z})\,(Q_1 \mid Q_2 \mid u_1[R] \mid \cdots \mid u_n[R])\,]$$

$$x^{\eta_1}\{\vec{u}\}.P \mid y[(\boldsymbol{\nu}\,\vec{z})(\overline{x}^{\eta_2}\{v\}.Q_1 \mid v[R] \mid Q_2)] \rightarrow P \mid u_1[R] \mid \cdots \mid u_n[R] \mid y[\,(\boldsymbol{\nu}\,\vec{z})\,(Q_1 \mid Q_2)\,]$$

where $fv(R) \cap \vec{z} = \varnothing$, $x \notin \vec{z}$, and $\mathsf{synch}_y(\eta_1, \eta_2)$ holds true.

Fig. 2. Reduction rules

The non-local rules are parametric in synch: different remote interaction patterns are obtained according whether synch is replaced by synch^S (shared channels), or synch^L (located channels).

The first rule describes local communication, which is exactly the same as in the polyadic π-calculus. The second rule describes the communication of a tuple $\vec{v}$ from a parent to its child y, which takes place provided that η_1 and η_2 and y match a synchronisation pattern, channel x is not locally restricted (i.e., $x \notin \vec{z}$), and no communicated variable is captured (i.e., $\vec{v} \cap \vec{z} = \varnothing$). The third rule is where extrusion of local restrictions of communicated variables takes place, as it corresponds to the case where a child y communicates to its parent a vector $\vec{v}$ of names. As for all remote synchronisations η_1 and η_2 and y must allow synchronisation and x must not be locally restricted (i.e., $x \notin \vec{z}$). Local (in y) restrictions of variables that are communicated to the parent (i.e., the variables in $\vec{v} \cap \vec{z}$) are extruded while the restrictions of the other variables (i.e., the variables in $\vec{z} \setminus \vec{v}$) stay in y.

The fourth rule states that in local mobility the body of the seal specified by the send action is copied as many times as specified by the receive action. This allows an easy implementation of operations like the copy of a seal $(\mathsf{copy}\,x\,\mathsf{as}\,z).P \stackrel{def}{=} (\boldsymbol{\nu}\,y)\,(\,\overline{y}^*\{x\} \mid y^*\{x\,,z\}.P\,)$ and its destruction $(\mathsf{destroy}\,x\,).P \stackrel{def}{=} (\boldsymbol{\nu}\,y)\,(\,\overline{y}^*\{x\} \mid y^*\{\ \}.P\,)$. The fifth rule states that a seal can be moved inside a child y provided that (i) η_1 and η_2 are y-corresponding locations, (ii) channel x is not locally restricted (i.e., $x \notin \vec{z}$), and (iii) no variable free in the moved process is captured (i.e., $fv(R) \cap \vec{z} = \varnothing$).

The last rule breaks the analogy between communication and mobility rules, and differs from semantics given in [14], as no extrusion is performed. In fact, the last rule requires that the body of the moved seal does not contain free any locally restricted variable (i.e., $fv(R) \cap \vec{z} = \varnothing$). This implies that all variables free in an exiting seal must already be known by the parent, either because they are non-local or because they were previously communicated to it. There are two reasons for choosing this more restrictive solution. First, this approach requires that private names are explicitly exported, giving the programmer a tighter control on local resources. Second, in a perspective implementation locally restricted channels would correspond to local variables. Thus, in case of mobility the free variables are handles that can be accessed only if some explicit reference is passed along with them. What we require here to be explicit, would be in any case implicit in the implementation.

3 Equivalences

In this section we study a semantic equivalence theory for the Seal Calculus. The goal is to determine what an 'adequate' semantic equivalence relation for agents should be. For example in [2,3] Cardelli and Gordon introduce and study a Morris-style contextual equivalence for Mobile Ambients according to which the process $(\boldsymbol{\nu}\, n)\, n[\, P\,]$ cannot be distinguished from the inactive process $\mathbf{0}$ when n does not occur free in P. The intuition is that since the name n is unknown both inside and outside the process, no other ambient can exert a capability on it. Thus it is as if the ambient n did not exist. This is summarised by the so-called *perfect firewall equation* which states that if $n \notin fv(P)$, then $(\boldsymbol{\nu}\, n)\, n[\, P\,] \simeq \mathbf{0}$. One may wonder whether this firewall is so perfect. Indeed the equation above does not ensure that n will not have any interaction with the surrounding context. As a matter of fact, n can enter another ambient that runs in parallel or exit the ambient it resides in. In other words n has total mobility freedom. More formally this means that if for example we consider the *commitment semantics* defined for Mobile Ambients in [4], then the process $(\boldsymbol{\nu}\, n)\, n[\, P\,]$ may emit actions such as $\mathtt{in}\, m$ and $\mathtt{out}\, m$.[2] This means that no reasonable bisimilarity relation that observes mobility capabilities will equate $\mathbf{0}$, that does not emit anything, with $(\boldsymbol{\nu}\, n)\, n[\, P\,]$. It is thus legitimate to wonder about the adequacy of the observation used to define $\simeq$.

A first answer to the question of what an appropriate notion of equivalence should be has been recently proposed for Ambients by Merro and Hennessy in a work [8] that strives towards our same goals and from which this section is deeply inspired. Merro and Hennessy work starts from Sangiorgi's observation in [12] that the algebraic theory of Ambients is poor. The goal of [8] is thus to modify Mobile Ambients so to endow them with an equational theory that is (*i*) richer, (*ii*) reasonable, (*iii*) adequate, and (*iv*) practicable. What do these four properties mean? Richer: that it proves equivalences more interesting than the simple structural congruence relation; reasonable: that it is a contextual equivalence that preserves reductions and some simple observational property; adequate: that it is invariant to different choices of observations (technically, of *barbs*); practicable: that it can be expressed in terms of bisimulation, whose coinductive nature ensures the existence of powerful proof techniques.

A first step in this direction was done by Levi and Sangiorgi [7] who extended Ambients by coactions. A reduction takes place only if an action synchronises with a corresponding coaction, which yields to a more satisfactory equational

[2] More precisely, according to the system in [4] a process of the form $(\boldsymbol{\nu}\, n)\, n[\, P\,]$ may emit $\overline{\mathit{enter}\ m}$ (and thus enter in a sibling ambient m) and $\mathit{exit}\ m$ (and thus exit from a surrounding ambient m).

theory. Nevertheless we are once more in the presence of a contextual equivalence which does not enjoy the last two properties. In [8] Merro and Hennessy extend (and modify) the work of [7] by adding to Ambients, besides coactions, also some *passwords*: an action and the corresponding coaction synchronise only if they possess the same password. Then, Merro and Hennessy define a bisimulation-based equivalence that is invariant for a large choice of observations. In other terms, they show that their extension enjoys the four required properties.

It is quite interesting to notice that all these modifications, proposed in order to endow Mobile Ambients with more sensible equational theories, make it closer and closer to the Seal Calculus: [7] requires mobility to be the consequence of a process synchronisation; [8] simply requires that the mobility takes place on channels (as Merro and Hennessy's passwords can be easily assimilated to channels)[3]. The very last step that distinguishes these Ambient variations from Seal is that Seal uses objective mobility — the agent is sent by the surrounding environment — while in Ambient-based calculi mobility is subjective — the agent sends itself — (as an aside, note that objective moves have also been added to Ambients by Cardelli, Ghelli and Gordon [1] in order to have more refined typings). So it seems quite natural that results similar to those of Merro and Hennessy can be stated for Seals without requiring any modification of its definition. This is what we do in this section, which constitutes the technical core and the difficult part of this work. Thus we start to define in Section 3.1 a labelled transition system and prove its equivalence with the reduction semantics of the previous section. Then in Section 3.2 we define a contextual equivalence (a barbed congruence) and a bisimilarity relation based on the previous labelled transition systems. We prove that the bisimilarity is a congruence and is sound with respect to (i.e. contained in) the contextual equivalence. So we have a notion of equivalence that nearly satisfies the four requirement we stated. To have the same results as in [8] it remains to prove the completeness of the bisimilarity. Unfortunately this seems to require non-trivial modifications as we explain at the end of the presentation.

3.1 Labelled Transition System

If we compare the study of equivalences for the Seal Calculus with the one done by Merro and Hennessy for the Ambient Calculus, then Seal presents two main difficulties. First, and foremost, the use of objective, rather than subjective, moves requires a three-party synchronisation (like in [6]) that introduces further complexity as it requires some intermediate ad hoc transitions. Second, the presence of channelled synchronisations together with the stricter discipline of Seal on private names make the handling of extrusion much more difficult.

[3] Merro and Hennessy also modify Levi and Sangiorgi's calculus so that the coaction of an `out` must be placed exactly as a receive action in Seal.

For the rest of this section we focus on the *shared* version of Seal. In particular the lts and the Definition 6 is sensible only for shared channels as some modifications are needed to account for the located variant.[4]

In Figure 4 we report the labelled transition system (lts) for the *Shared* Seal.

Labels			**Activities**			**Locations**	
ℓ ::=	τ	internal action	a ::=	$x^\eta(\vec{y})$	input	γ ::=	$*$ here
|	P_z	seal freeze	|	$\overline{x}^\eta(\vec{y})$	output	|	z inside z
|	P^z	seal chained	|	$\overline{x}^\eta\{y\}$	send		
|	$\gamma[a]$	activity a at γ	|	$\overline{x}^\eta\{P\}$	capsule		
			|	$x^\eta\{P\}$	receive		
			|	x^η_z	lock		

The free names of a label, $fv(\ell)$, are defined according to the following rules:

$$fv(\tau) = \varnothing \qquad fv(P_z) = fv(P^z) = \{z\} \cup fv(P) \qquad fv(\gamma[a]) = fv(\gamma) \cup fv(a)$$
$$fv(x^\eta(\vec{y})) = fv(\overline{x}^\eta(\vec{y})) = \{x, \vec{y}\} \cup fv(\eta) \qquad fv(\overline{x}^\eta\{y\}) = \{x, y\} \cup fv(\eta)$$
$$fv(\overline{x}^\eta\{P\}) = fv(x^\eta\{P\}) = \{x\} \cup fv(\eta) \cup fv(P) \qquad fv(x^\eta_z) = \{x, z\} \cup fv(\eta)$$

The label τ is the standard silent label indicating internal synchronisation. The label P_z denotes a seal z running P that *freezes* itself, in order to be moved. The label P^z denotes a partial synchronisation: a process willing to move a seal named z and a process willing to receive a seal with body P synchronised, and are now looking for the frozen seal to be moved. An activity $\gamma[a]$ denotes the offer of a visible interaction from a process located at γ.

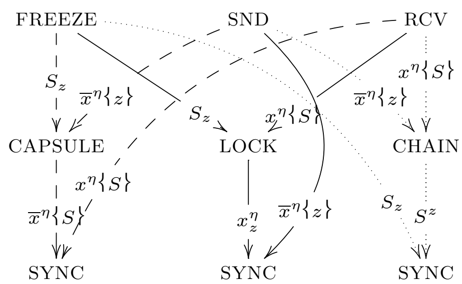


Fig. 3. Synchronisation paths.

More in detail, the $x^\eta(y)$ label denotes the offer of the input of the value y over channel x tagged by η, the $\overline{x}^\eta(y)$ label denotes the offer of the output of the value y over channel x tagged by η, the $\overline{x}^\eta\{y\}$ label denotes the offer of sending a seal named y over channel x tagged by η, and the $x^\eta\{P\}$ label denotes the offer of receiving the seal body P over channel x tagged by η. The label $\overline{x}^\eta\{P\}$ represents the action of *serialising* a seal: its emission indicates that a process willing to send a seal over a channel found it, serialised it, and is now waiting for synchronising with a receiver process. The label x^η_z, too, denotes a partial synchronisation: a process willing to receive a seal at x^η synchronised with the corresponding frozen seal of name z, and is now looking for a sender. If γ is $*$, then the activity takes place at the current level, if γ is a name z then the activity takes place inside a seal named z. Not all activities are visible outside the current seal, and none is visible outside the containing seal. A schema describing the possible synchronisation paths for mobility is reported in Figure 3 (localities

[4] In *Located* Seal the two subprocesses in $x^*(y) \mid (\boldsymbol{\nu}\, a)\; a[\,\overline{x}^\uparrow(z)\,]$ can synchronise causing the extrusion of $(\boldsymbol{\nu}\, a)$, while in *Shared* Seal they cannot.

and communications have been omitted for clarity). The $\curlyvee$ relation is used to keep track of the couple of labels that match to generate a τ transition.

Definition 3. *Let $\curlyvee$ be the smallest binary* symmetric *relation on labels containing the following relation:*

$$\begin{aligned}&\{\ (\ \gamma_1[\overline{x}^{\eta_1}(\vec{y})]\ ,\ \gamma_2[x^{\eta_2}(\vec{y})]\ )\mid \mathcal{M}\ \}\ \cup\ \{\ (\ \gamma_1[\overline{x}^{\eta_1}\{\!\!\{S\}\!\!\}]\ ,\ \ \gamma_2[x^{\eta_2}\{\!\!\{S\}\!\!\}]\ )\mid \mathcal{M}\ \}\\ &\cup\ \{\ (\ \gamma_1[x_z^{\eta_1}]\ ,\ \gamma_2[\overline{x}^{\eta_2}\{\!\!\{z\}\!\!\}]\ )\mid \mathcal{M}\ \}\ \cup\ \{\ (\ S_z\ ,\ S^z\ )\ \}\end{aligned}$$

where $\mathcal{M} \stackrel{def}{=} (\gamma_1 = \eta_1 = \gamma_2 = \eta_2 = *) \vee (\gamma_1 = * \ \wedge\ \mathsf{synch}_{\gamma_2}(\eta_1, \eta_2)) \vee (\gamma_2 = * \ \wedge\ \mathsf{synch}_{\gamma_1}(\eta_2, \eta_1))$.

The labelled transition relation has the form $A \vdash P \xrightarrow{\ell} P'$ where A is a finite set of names and $fv(P) \subseteq A$; it has to be read as 'in a state where names in A may be known by process P and by its environment, the process P can perform ℓ and become P''. This presentation, borrowed from [13], allows us to drop many side conditions dealing with the extrusion of names. The lts is reported in Figure 4. It defines an *early* semantics, as rules (IN) and (RCV) show. This avoids explicitly dealing with process substitutions and is well suited to study bisimulation. The conditions $\gamma = * \Rightarrow \eta \neq \uparrow$ on rule (OPEN CAPSULE) and $fv(S) \subseteq A$ on rule (SEAL LABEL) guarantee that moving a seal body outside the current seal cannot extrude a local name.

The following theorem states the equivalence between the lts and the semantics in chemical style.

Theorem 1. *Let P be a process: (i) if $fv(P) \subseteq A$ and $A \vdash P \xrightarrow{\tau} Q$, then $P \rightarrow Q$, and (ii) if $P \rightarrow Q$ then there exists $A \supseteq fv(P)$ such that $A \vdash P \xrightarrow{\tau} Q'$, where $Q' \equiv Q$.*

3.2 Equivalence relations

We next define a contextual equivalence for Seal processes. This equivalence is based on the observation of the presence at top level of a seal whose name is unrestricted. Such an observation, due to Cardelli and Gordon [2,3], can be interpreted as the ability of the top-level process to interact with that seal.

We write $\rightarrow^*$ and $\Rightarrow$ for the reflexive and transitive closure of $\rightarrow$ and $\xrightarrow{\tau}$, respectively. We have the following definitions:

Definition 4 (Barbs). *We write $P \downarrow n$ if and only if there exist Q, R, $\vec{x}$ such that $P \equiv (\boldsymbol{\nu}\, \vec{x})\, (n[\,Q\,] \mid R)$ where $n \notin \vec{x}$. We write $P \Downarrow n$ if there exists P' such that $P \rightarrow^* P'$ and $P' \downarrow n$.*

Definition 5 (Barbed Congruence). *Barbed congruence $\cong$ is the largest congruence relation over processes that (i) is reduction closed, that is: if $P \cong Q$ and $P \rightarrow P'$, then there exists Q' such that $Q \rightarrow^* Q'$ and $P' \cong Q'$; (ii) preserves barbs, that is: $P \cong Q$ and $P \downarrow n$ implies $Q \Downarrow n$.*

Congruence

(PAR)
$$\frac{A \vdash P \xrightarrow{\ell} P'}{A \vdash P \mid Q \xrightarrow{\ell} P' \mid Q}$$

(RES) $\forall i,\ x_i \notin fv(\ell)$
$$\frac{A \cdot \vec{x} \vdash P \xrightarrow{\ell} P'}{A \vdash (\boldsymbol{\nu}\, \vec{x})\, P \xrightarrow{\ell} (\boldsymbol{\nu}\, \vec{x})\, P'}$$

(BANG)
$$\frac{}{A \vdash !\gamma.P \xrightarrow{\gamma} P \mid !\gamma.P}$$

(OPEN COM) $y, \eta, \gamma \notin \vec{u}$
$$\frac{A \cdot \vec{u} \vdash P \xrightarrow{\gamma[\overline{y}^{\eta}(\vec{x})]} P'}{A \vdash (\boldsymbol{\nu}\, \vec{u})\, P \xrightarrow{\gamma[\overline{y}^{\eta}(\vec{x})]} (\boldsymbol{\nu}\, \vec{u} \setminus \vec{x})\, P'}$$

(OPEN FREEZE) $z \notin \vec{u}$
$$\frac{A \cdot \vec{u} \vdash P \xrightarrow{S_z} P'}{A \vdash (\boldsymbol{\nu}\, \vec{u})\, P \xrightarrow{S_z} (\boldsymbol{\nu}\, \vec{u} \setminus fv(S))\, P'}$$

(SEAL TAU)
$$\frac{A \vdash P \xrightarrow{\tau} P'}{A \vdash x[\,P\,] \xrightarrow{\tau} x[\,P'\,]}$$

(OPEN CAPSULE) $y, \eta, \gamma \notin \vec{u}$; if $\gamma = *$ then $\eta \neq \uparrow$
$$\frac{A \cdot \vec{u} \vdash P \xrightarrow{\gamma[\overline{y}^{\eta}\{\!\!\{S\}\!\!\}]} P'}{A \vdash (\boldsymbol{\nu}\, \vec{u})\, P \xrightarrow{\gamma[\overline{y}^{\eta}\{\!\!\{S\}\!\!\}]} (\boldsymbol{\nu}\, \vec{u} \setminus fv(S))\, P'}$$

(SEAL LABEL) $fv(S) \subseteq A, \exists \eta'.\mathsf{synch}_x(\eta', \eta)$
$$\frac{A \vdash P \xrightarrow{*[a]} P' \quad a \in \{y^{\eta}(\vec{z}), \overline{y}^{\eta}(\vec{z}), y^{\eta}\{\!\!\{Q\}\!\!\}, \overline{y}^{\eta}\{\!\!\{S\}\!\!\}\}}{A \vdash x[\,P\,] \xrightarrow{x[\,a\,]} x[\,P'\,]}$$

Communication

(OUT)
$$\frac{}{A \vdash \overline{x}^{\eta}(\vec{y}).P \xrightarrow{*[\overline{x}^{\eta}(\vec{y})]} P}$$

(IN)
$$\frac{}{A \vdash x^{\eta}(\vec{y}).P \xrightarrow{*[x^{\eta}(\vec{v})]} P\{^{\vec{v}}/_{\vec{y}}\}}$$

Mobility

(SND)
$$\frac{}{A \vdash \overline{x}^{\eta}\{\!\!\{y\}\!\!\}.P \xrightarrow{*[\overline{x}^{\eta}\{\!\!\{y\}\!\!\}]} P}$$

(RCV)
$$\frac{}{A \vdash x^{\eta}\{\!\!\{\vec{y}\}\!\!\}.P \xrightarrow{*[x^{\eta}\{\!\!\{Q\}\!\!\}]} P \mid y_1[\,Q\,] \mid \cdots \mid y_n[\,Q\,]}$$

(CAPSULE)
$$\frac{A \vdash P \xrightarrow{S_z} P' \quad A \vdash Q \xrightarrow{*[\overline{x}^{\eta}\{\!\!\{z\}\!\!\}]} Q'}{A \vdash P \mid Q \xrightarrow{*[\overline{x}^{\eta}\{\!\!\{S\}\!\!\}]} P' \mid Q'}$$

(LOCK) $\gamma = \eta = *$ or $\exists \eta'.\mathsf{synch}_\gamma(\eta', \eta)$
$$\frac{A \vdash P \xrightarrow{S_z} P' \quad A \vdash Q \xrightarrow{\gamma[x^{\eta}\{\!\!\{S\}\!\!\}]} Q'}{A \vdash P \mid Q \xrightarrow{\gamma[x_z^{\eta}]} (\boldsymbol{\nu}\, fv(S) \setminus A)\,(P' \mid Q')}$$

(FREEZE)
$$\frac{}{A \vdash x[\,P\,] \xrightarrow{P_x} \mathbf{0}}$$

(CHAIN) $\gamma = \eta_1 = \eta_2 = *$ or $\mathsf{synch}_\gamma(\eta_1, \eta_2)$
$$\frac{A \vdash P \xrightarrow{*[\overline{x}^{\eta_1}\{\!\!\{y\}\!\!\}]} P' \quad A \vdash Q \xrightarrow{\gamma[x^{\eta_2}\{\!\!\{S\}\!\!\}]} Q'}{A \vdash P \mid Q \xrightarrow{S^y} P' \mid Q'}$$

Synchronisation

(SYNC) $\ell_1 \curlyvee \ell_2$
$$\frac{A \vdash P \xrightarrow{\ell_1} P' \qquad A \vdash Q \xrightarrow{\ell_2} Q'}{A \vdash P \mid Q \xrightarrow{\tau} (\boldsymbol{\nu}\,(fv(\ell_1) \cup fv(\ell_2)) \setminus A)\,(P' \mid Q')}$$

The symmetric rules for (PAR), (CAPSULE), (LOCK), and (CHAIN) are omitted.
Notation: $A \cdot \vec{u}$ is defined as $A \cup \vec{u}$ if A and $\vec{u}$ are disjoint, it is undefined otherwise.

Fig. 4. Labelled transition system for shared channels.

As a first application of these definitions we can show that $P = (\boldsymbol{\nu}\, x)\, n[\, R\,]$ is not equivalent to $Q = n[\,(\boldsymbol{\nu}\, x)\, R\,]$, and prove in this way the unsoundness of rule $(*)$ given in Section 2. It just suffices to take $R = \overline{y}^{\uparrow}(x) \mid x^{\uparrow}()$ and to consider the context $\mathcal{C}[-] = \mathsf{copy}\, n\, \mathsf{as}\, m.y^{n}(u).\overline{u}^{m}().b[\,] \mid [-]$, where b is fresh. Then $\mathcal{C}[P] \rightarrow^{*} P'$ and $P' \downarrow b$ while there is no Q' such that $\mathcal{C}[Q] \rightarrow^{*} Q'$ and $Q' \Downarrow b$.

As the above example shows, contextual equivalence is useful to prove that two processes are *not* equivalent (it suffices to find a context that differentiate them) but it is unfit to prove the equivalence of processes. To that end we seek for a coinductive characterisation of the barbed congruence above.

First of all, remark that the exposure of a barb corresponds to the emission of a (FREEZE) label in the lts:

Lemma 1. *$P \downarrow n$ iff $A \vdash P \xrightarrow{Q_n} P'$ for some P', Q, and A, with $fv(P) \subseteq A$.*

The lemma above shows that the observation used in the contextual equivalence is insensitive to the particular process Q occurring in the label of the labelled transition. Thus, we expect a coinductive characterisation of this equivalence not to be strict in matching (higher-order) labels in which processes occur. As a matter of facts, when agents can be communicated, requiring processes appearing in matching labels to be equal is overly restrictive (as, for instance, in our case $x[\,\mathbf{0}\,]$ and $x[\,y^{z}()\,]$ would then not be equivalent). On the other hand requiring them to be bisimilar is source of problems when agent mobility requires extrusions of names.

To escape this *impasse* we resort to the intuition underlying the definition of Sangiorgi's *delay bisimilarity* for HOπ ([10], [11]), and require that the outcomes of two bisimilar processes emitting higher order transitions are equivalent with respect to every possible interaction with a context. To that end we introduce the definition of *receiving contexts*. These are processes parametric in two processes X and Y, where Y may get replicated. Receiving contexts represents all the possible outcomes that may result from the migration of a seal, where the parameter processes X and Y stand, respectively, for the residuum of the process that sent the seal and for the body of the moved seal.

Definition 6 (Receiving Context). *Given two processes X and Y where $fv(X) \subseteq A$, a* receiving context *$\mathcal{D}^{A}_{\gamma,\eta}[X, Y]$ and its* associated environment *$A_{\mathcal{D}^{A}_{\gamma,\eta}[X,Y]}$ are respectively a process and a context defined as:*

if $\gamma, \eta = *, *$, or $\gamma, \eta = z, \uparrow$ and $fv(Y) \subseteq A$, then for all $\vec{z}$ such that $fv(\mathcal{D}^{A}_{\gamma,\eta}[X, Y]) \subseteq A \cup \vec{z}$

$$(\boldsymbol{\nu}\, fv(Y) \setminus A)\,(\; X \mid z_1[\,Y\,] \mid \cdots \mid z_n[\,Y\,]\;)$$

and its associated environment $A_{\mathcal{D}^{A}_{*,*}[X,Y]}$ is $A \cup \vec{z}$;
if $\gamma, \eta = *, z$, then for all $\vec{v}, \vec{z}$, and U such that $fv(Y) \cap \vec{v} = \varnothing$, and $fv(\mathcal{D}^{A}_{*,z}[X, Y]) \subseteq A \cup (\vec{z} \setminus \vec{v})$

$$(\boldsymbol{\nu}\, fv(Y) \setminus A)\,(\; X \mid z[\,(\boldsymbol{\nu}\, \vec{v})\,(\; z_1[\,Y\,] \mid \cdots \mid z_n[\,Y\,] \mid U\;)\,]\;)$$

and its associated environment $A_{\mathcal{D}^{A}_{*,z}[X,Y]}$ is $A \cup (\vec{z} \setminus \vec{v})$;

if $\gamma, \eta = *, \uparrow$, then for all $\vec{v}, U, z$, such that $fv(Y) \cap \vec{v} = \varnothing$, $fv(Y) \subseteq A$, and $fv(\mathcal{D}^A_{\gamma,\uparrow}[X, Y]) \subseteq (A \setminus \vec{v}) \cup \vec{z}$

$$z[\,(\boldsymbol{\nu}\,\vec{v})\,(X \mid U)\,] \mid z_1[\,Y\,] \mid \cdots \mid z_n[\,Y\,]$$

and its associated environment $A_{\mathcal{D}^A_{\gamma,\uparrow}[X,Y]}$ is $(A \setminus \vec{v}) \cup \vec{z}$.[5]

We write $\mathcal{D}^A[X, Y]$ when we quantify over all γ, η and abbreviate $A_{\mathcal{D}^A_{\gamma,\eta}[X,Y]}$ by $A_{\mathcal{D}}$ when no ambiguity arises.

Receiving contexts are then used to compare higher-order labelled transitions:

Definition 7 (Hoe Bisimilarity).

Let hoe bisimilarity $\sim$ be the largest family of symmetric relations indexed by finite sets of names such that each $\sim_A$ is a binary relation over $\{P \mid fv(P) \subseteq A\}$ and for all $P \sim_A Q$ the following conditions hold:

1. *if $A \vdash P \xrightarrow{\tau} P'$ then there exists a Q' such that $A \vdash Q \Rightarrow Q'$ and $P' \sim_A Q'$;*
2. *if $A \vdash P \xrightarrow{\ell} P'$ and $\ell \in \{\ \gamma[x^\eta(\vec{y})], \gamma[\overline{x}^\eta(\vec{y})], *[\overline{x}^\eta\{\!|y|\!\}], \gamma[x^\eta\{\!|S|\!\}], S^z, \gamma[x^\eta_z]\ \}$, then there exists a Q' such that $A \vdash Q \Rightarrow\!\xrightarrow{\ell} Q'$ and $P' \sim_{A \cup fv(\ell)} Q'$;*
3. *if $A \vdash P \xrightarrow{R_z} P'$ then there exist Q', S such that $A \vdash Q \Rightarrow\!\xrightarrow{S_z} Q'$ and for all admissible contexts $\mathcal{D}^A[-,-]$ it holds $\mathcal{D}^A[P', R]\ \sim_{A_{\mathcal{D}}}\ \mathcal{D}^A[Q', S]$;*
4. *if $A \vdash P \xrightarrow{\gamma[\overline{x}^\eta\{\!|R|\!\}]} P'$ then there exist Q', S such that $A \vdash Q \Rightarrow\!\xrightarrow{\gamma[\overline{x}^\eta\{\!|S|\!\}]} Q'$ and for all admissible contexts $\mathcal{D}^A_{\gamma,\eta}[-,-]$ it holds $\mathcal{D}^A_{\gamma,\eta}[P', R]\ \sim_{A_{\mathcal{D}}}\ \mathcal{D}^A_{\gamma,\eta}[Q', S]$;*
5. *for all substitutions σ, $P\sigma \sim_{A\sigma} Q\sigma$;*

where a context $\mathcal{D}^A[-,-]$ is admissible *if both process substitutions $\mathcal{D}^A[P', R]$ and $\mathcal{D}^A[Q', S]$ are well-formed (i.e. no name capture arises).*

The first two cases of Definition 7 handle all low-order labels, as well as labels originating from receive actions: these do not deserve a special treatment because our early semantics implicitly tests all possible interactions. The cases 3. and 4. check mobility, by testing against all possible outcomes after a mobility interaction with a context.

The most important result of this work is the soundness of bisimilarity:

Theorem 2 (Soundness). *Hoe bisimilarity is sound with respect to barbed congruence: if $P \sim_A Q$ for some A, then $P \cong Q$.*

The proof of this theorem is a consequence of the following lemma.

Lemma 2. *Hoe bisimilarity is a congruence.*

As an application of this theory let us go back to the perfect firewall equation at the beginning of this section. It is quite easy to prove that $(\boldsymbol{\nu}\,x)\,x[\,P\,] \sim \mathbf{0}$ as it suffices to exhibit the following bisimulation $\mathcal{B} = \bigcup_A \mathcal{B}_A$, where $\mathcal{B}_A = \{((\boldsymbol{\nu}\,x)\,x[\,Q\,], \mathbf{0}) \mid P \rightarrow^* Q\} \cup \{(\mathbf{0}, (\boldsymbol{\nu}\,x)\,x[\,Q\,]) \mid P \rightarrow^* Q\}$ if $fv(P) \subseteq A$, and empty otherwise. The use of hoe bisimilarity ensures us that this firewall *is* perfect, as it cannot emit anything but the silent label. The soundness of bisimilarity implies $(\boldsymbol{\nu}\,x)\,x[\,P\,] \cong \mathbf{0}$.

[5] Note that the associated environment is defined in term of the process rather than of the context, as its definition depends on the possibly fresh variables $\vec{z}$.

Open issues. The definition of hoe bisimilarity for located channels does not seem worth to be pursued. It is easy to see that with localised channels hoe bisimilarity is not a congruence (the problem being the extrusion of seal names corresponding to $x[a]$ labels), and while it is not difficult to make the needed modifications, this does not seem interesting since the resulting equivalence would be too strong (e.g., $(\boldsymbol{\nu}\, x)\, x[\,P\,] \not\approx \mathbf{0}$).

A more interesting problem is that Hoe bisimilarity is not complete with respect to barbed congruence. Although is is easy to construct a counterexample by exploiting the fact that weak transitions $\Rightarrow\xrightarrow{\ell}$ do not allow τ moves a visible action, the problem is deeper than that. First, since Seal Calculus is an extension of the π-calculus and in the π-calculus the matching operator is necessary to completeness, then this same operator may be required also in Seal; second, in the three party synchronisation the intermediate actions are not observable, therefore it seems quite hard to find a context to separate them. So completeness cannot be easily reached and needs much more research effort.

References

1. L. Cardelli, G. Ghelli, and A. D. Gordon. Ambient groups and mobility types. In *Intl. Conf. IFIP TCS*, number 1872 in LNCS, pages 333–347. Springer, 2000.
2. L. Cardelli and A. Gordon. Mobile Ambients. In *Proceedings of F0SSaCS'98*, number 1378 in Lecture Notes in Computer Science, pages 140–155. Springer, 1998.
3. L. Cardelli and A. Gordon. Equational properties for Mobile Ambients. In *Proceedings FoSSaCS '99*. Springer LNCS, 1999.
4. L. Cardelli and A. Gordon. A commitment relation for the Ambient Calculus. Available at http://research.microsoft.com/~adg/Publications/ambient-commitment.pdf, 2000.
5. G. Castagna, G. Ghelli, and F. Zappa Nardelli. Typing mobility in the Seal Calculus. In *12th. International Conference on Concurrency Theory*, number 2154 in Lecture Notes in Computer Science, pages 82–101, Springer, 2001.
6. J.C. Godskesen, T. Hildebrandt, and V. Sassone. A calculus of mobile resources. In *CONCUR 2002*, Lecture Notes in Computer Science. Springer, 2002.
7. F. Levi and D. Sangiorgi. Controlling interference in Ambients. In *POPL '00*, pages 352–364. ACM Press, 2000.
8. M. Merro and M. Hennessy. Bisimulation conguences in Safe Ambients. In *Proc. of the 29th ACM Symp. on Principles of Programming Languages*. ACM Press, 2002.
9. R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, Parts I and II. *Information and Computation*, 100:1–77, September 1992.
10. D. Sangiorgi. *Expressing Mobility in Process Algebras: First-Order and Higher-Order Paradigms.* PhD thesis CST–99–93, University of Edinburgh, 1992.
11. D. Sangiorgi. Bisimulation for Higher-Order Process Calculi. *Information and Computation*, 131(2):141–178, 1996.
12. D. Sangiorgi. Extensionality and intensionality of the ambient logic. In *Proc. of the 28th ACM Symp. on Principles of Programming Languages*, London, 2001.
13. P. Sewell and J. Vitek. Secure composition of insecure components. In *12th IEEE Computer Security Foundations Workshop*, 1999.
14. J. Vitek and G. Castagna. Seal: A framework for secure mobile computations. In *Internet Programming Languages*, number 1686 in Lecture Notes in Computer Science, pages 47–77. Springer, 1999.