A Filter Model for Safe Ambients Calculus

I. Margaria M. Zacchi
Dipartimento di Informatica – Università di Torino

1 Introduction

The *Ambient Calculus* (AC) [1] is a process calculus for describing mobile computations. Unit of the movement is the ambient n[P] that represents a bounded space named n in which the process P can make computations, exchange messages or exercise movement capabilities: i.e. enter or exit other named ambients or dissolve ambient boundaries. An interesting topic in Ambient Calculus is the study of an appropriate notion of semantics equivalence and of the methods for establishing such equivalences [2]; the principal equivalence relation proposed for the Ambient Calculus is a contextual equivalence based on the observability of ambients $\approx_{obs}$. In [3] the equivalence between terms of Ambient Calculus has been studied by means a filter model, that results to be fully abstract with respect to the contextual equivalence $\approx_{obs}$. The model is designed via a type system, where types represent properties of processes. In this paper we extend the definition of the filter model to a variant of Ambient Calculus: the Safe Ambients (SA) Calculus. As usual in filter models, the processes of SA are interpreted as the set of their types. The inclusion relation between these sets of properties induces an ordering $\subseteq_F$ on processes. We prove the adequacy of the model: i.e.

$P \subseteq_F Q \Rightarrow P \subseteq_{obs} Q$.

2 The Calculus

The calculus we consider here is basically the *Safe Ambients Calculus* [4] in which every capability has a co-capability. We omit communication and replication.
The syntax of calculus is given in Table 1. As usual ≡ denotes *structural congruence*, Table 2 shows the *reduction rules*, representing the behavior of processes. Notice that for the out-reduction rule we follow the variant proposed in [5].

Table 1

Names: $n \in \mathcal{N}$

Capabilities: $c \in \mathcal{C}$

$c ::= in\ n \mid out\ n \mid open\ n \mid co\text{-}in\ n \mid co\text{-}out\ n \mid co\text{-}open\ n$

Processes: $P \in \mathcal{P}$

$P ::= 0 \mid c.P \mid P_1 \mid P_2 \mid n[P] \mid (\nu n)P$

Table 2

→ *is the least equivalence relation that* :
i) is preserved by all operators except prefixing
ii) satisfies the rules above:

- $m[in\ n.P \mid Q] \mid n[co\text{-}in\ n.R \mid S] \rightarrow n[m[P \mid Q] \mid R \mid S]$ (Red-In)
- $m[n[out\ m.P \mid Q] \mid R] \mid co\text{-}out\ m.S \rightarrow n[P \mid Q] \mid m[R] \mid S$ (Red-Out)
- $open\ n.P \mid n[co\text{-}open\ n.Q \mid R] \rightarrow P \mid Q \mid R$ (Red-Open)
- $P \equiv Q\ ,\ Q \rightarrow R,\ R \equiv S$ implies $P \rightarrow S$ (Red-Struct)

$\rightarrow^*$ *is the reflexive and transitive closure of* →.

Two processes P and Q are considered equivalent if, placed in arbitrary contexts, exhibit the same ambients; formally ([5]):

Observational Equivalence Definition (i) A process P *exhibits an ambient* n: $P \Downarrow n$ if

$P \rightarrow^* (\nu\tilde{m})(n[\textit{co-open}\ n.Q|R]|S)$ for some processes Q, R, S $(n \notin \tilde{m})$.

(ii) $P \subseteq Q$ if for all contexts $C[\ ]$ and ambients n : $C[P] \Downarrow n \Rightarrow C[Q] \Downarrow n$.

(iii) $P \approx Q$ if $P \subseteq Q$ and $Q \subseteq P$.

3 Types

Like in [3] types are intended to provide partial information about processes, giving their properties. We consider the mobility actions, the ambients and parallel composition. The formal definition of the set of types $\mathcal{T}$ is given in Table 3.

Table 3

Prefixes1:	$\mu ::= \textit{in}\ n \mid \textit{out}\ n \mid \textit{open}\ n \mid \textit{co-in}\ n \mid \textit{co-out}\ n \mid \textit{co-open}\ n \mid \textit{pop}_m\ n \mid \textit{free}\ n$
Prefixes2:	$\alpha ::= \textit{enter}_m\ n \mid \textit{exit}_m\ n \mid \textit{co-enter}\ n$
Actions:	$\gamma ::= \mu \mid \alpha$
Types:	$\sigma ::= \omega \mid \mu.\sigma \mid \alpha.(\nu\tilde{m})\ (<\sigma>_n \tau) \mid n[\sigma] \mid (\nu n)\,\sigma \mid \sigma \mid \tau \mid \sigma \wedge \tau$

Our definition of types is inspired both by the type definition of [3] and by the labelled transition system of [5]. In particular to [5] is due the idea of *action*, as extension of the original definition of capability. In fact each capability gives rise to an action (elementary action), but, when inserted in ambients, it can induce further actions; by way of example we can say that the process $k[\textit{in}\ n.P]$ has the capability to *enter the ambient* n, after that it has a continuation, that is expressed by the pair $(\nu\tilde{m})(<\sigma>_n \tau)$ (cfr. notion of *concretion* in [5]); a type $\alpha.(\nu\tilde{m})(<\sigma>_n \tau))$ models the behavior of a process that exercises the action α and then leaves inside the ambient n a process of type σ and, outside the ambient n, a process of type τ; $\tilde{m}$ represents the set of private names shared by σ and τ. The pairs ($\textit{enter}_m\ n$, *co-enter* n), ($\textit{pop}_m\ n$, *co-out* n), (*free* n, *open* n) are said *matching* pairs.

Type ω represents a property true for all processes, whereas the conjunction type constructor $\wedge$ is added to model nondeterminism: a process having type $\sigma \wedge \tau$ can possibly exhibit, in different reduction paths, both property σ and τ.

On the set of types $\mathcal{T}$ is defined a partial order relation $\leq$, representing *entailment*; $\sigma \leq \tau$ means that the property σ entails property τ; $\sigma \simeq \tau$ iff $\sigma \leq \tau$ and $\tau \leq \sigma$. Among the Type Entailment Rules, particular relevance has a sequentialization rule of kind:

$$\gamma_1.\sigma \mid \gamma_1.\tau \quad \simeq \quad \gamma_1.(\sigma \mid \tau) \wedge \gamma_2.(\sigma \mid \tau) \qquad \text{if } \gamma_1 \text{ and } \gamma_2 \text{ do not match}$$
$$\simeq \quad \sigma' \mid \tau' \wedge \gamma_1.(\sigma \mid \tau) \wedge \gamma_2.(\sigma \mid \tau) \qquad \text{if } \gamma_1 \text{ and } \gamma_2 \text{ match}$$

The application of this rule allows to translate a parallel composition of types into a nondeterministic choice between *sequential* types. A sequential type ϕ models the behavior of a process consisting of a sequence of actions, formally it is defined inductively in the following way: $\phi ::= \omega \mid \mu.\phi \mid \alpha.(\nu\tilde{m})(<\phi_1>_n \phi_2)$. Using this fact we can prove that every type has a *normal form*, consisting in a nondeterministic choice of sequential types.

Normal Form Lemma. For all $\sigma \in \mathcal{T}$ there is a unique type $\bigwedge_{i\in[1\ldots n]} \xi_i$, where ξ_i are sequential types, such that $\sigma \simeq \bigwedge_{i\in[1\ldots n]} \xi_i$. We call it the normal form of σ, denoted by $\textit{nf}(\sigma)$.

Types are associated with processes by means of type assignment rules, shown in Table 4.
We can prove that congruent processes have the same types and that types are preserved under subject expansion:

Subject Congruence Lemma. $P : \sigma$ and $P \equiv Q \Rightarrow Q : \sigma$.

Subject Expansion Lemma. $Q : \sigma$ and $P \rightarrow^* Q \Rightarrow P : \sigma$.

Table 4

$$(\omega)\quad P : \omega \qquad (\text{prefix})\quad \frac{P : \sigma \quad c \in C}{c.P : \mu_c.\sigma} \qquad (\text{amb})\quad \frac{P : \sigma \quad n \in N}{n[P] : n[\sigma]}$$

$$(|)\quad \frac{P_1 : \sigma \quad P_2 : \tau}{P_1 | P_2 : \sigma \mid \tau} \qquad (\text{res})\quad \frac{P : \sigma}{(\nu n)P : (\nu n)\sigma} \qquad (\leq)\quad \frac{P : \sigma \quad \sigma \leq \tau}{P : \tau}$$

$$(\wedge)\quad \frac{P : \sigma \quad P : \tau}{P : \sigma \wedge \tau}$$

4 The Filter Model

$\mathcal{F}(T)$ is the set of filters on the set of types T. If $A \subseteq T$ then $\uparrow A$ denotes the filter generated by A, obtained by closing A under finite intersection and by (upper closing A under) $\leq$. Let $par : \mathcal{F}(T) \times \mathcal{F}(T) \to \mathcal{F}(T)$ be the function defined by $par(F, G) = \uparrow\{\sigma \mid \tau \mid \sigma \in F \text{ and } \tau \in G\}$. The *interpretation* of a process is a function $[\![-]\!] : P \to \mathcal{F}(T)$, defined as follows:

- $[\![0]\!] = \uparrow\{\omega\}$
- $[\![c.P]\!] = \uparrow\{\mu_c.\sigma \mid \sigma \in [\![P]\!]\}$
- $[\![n[P]]\!] = \uparrow\{n[\sigma] \mid \sigma \in [\![P]\!]\}$
- $[\![P | Q]\!] = par([\![P]\!], [\![Q]\!])$
- $[\![(\nu n)P]\!] = \uparrow\{(\nu n)\sigma \mid \sigma \in [\![P]\!]\}$
-

The basic theorem of the filter models is that the interpretation of a term is given by the set of its types: $[\![P]\!] = \{\sigma \mid P : \sigma\}$

The inclusion on filters gives rise to an order relation $\subseteq_{\mathcal{F}}$ on processes, in the sense that $P \subseteq_{\mathcal{F}} Q$ if and only if $[\![P]\!] \subseteq [\![Q]\!]$

Adequacy

The proof of adequacy is done by defining an interpretation of types as set of processes and by proving that a process P belongs to the interpretation of the type σ if and only if σ belongs to the filter $[\![P]\!]$. To prove the adequacy of the model of the model it is essential to prove the soundness of type inclusion relation: $\sigma \leq \tau$ implies $[\![\sigma]\!] \subseteq [\![\tau]\!]$.

Soundness and Completeness Theorem $P : \sigma \Leftrightarrow P \in [\![\sigma]\!]$.

We can to characterize observational exhibition of ambients by means of typing:

Resource property. $P : \mathit{free}\ n.\omega \Leftrightarrow P \Downarrow n$

Adequacy Theorem. If $P \subseteq_{\mathcal{F}} Q$ then $P \subseteq Q$.

References

1. L. Cardelli, A.D. Gordon. Mobile Ambients. In *FoSSaCS* '98, volume 1378 of *LNCS*, pages 140-155, Berlin 1998. Springer Verlag.

2. L. Cardelli, A.D. Gordon. Equational Properties of Mobile Ambients. In *FoSSaCS* '99, volume 1578 of *LNCS*, pages 212-226, Berlin 1999. Springer Verlag**.**
3. M. Coppo, M. Dezani. A Fully Abstract Model for Higher-Order Mobile Ambients. In VMCAI '02, LNCS 2294, pages 255-271. Springer, 2002.
4. F. Levi, D. Sangiorgi. Controlling Interferences in Ambients. *Proc. 27th POPL*, ACM Press, 2000.
5. M. Merro, M. Hennessy. Bisimulation Congruences in Safe Ambients. *ACM SIGPLAN Notices,* 31(1), pages 71-80, 2002.

Completeness

To prove completeness we define, for every sequential type ϕ, a *Context Term* $C_\phi^n[\bullet]$, where n is an ambient name, fresh with respect to ϕ. The behavior of this terms is that for every process Q , for which n is fresh, $C_\phi^n[Q] \Downarrow n$ iff $Q : \phi$. □

Table 1

Names: $n \in \mathcal{N}$

Capabilities: $c \in \mathcal{C}$

$c ::= in\ n \mid out\ n \mid open\ n \mid co\text{-}in\ n \mid co\text{-}out\ n \mid co\text{-}open\ n$

Processes: $P \in \mathcal{P}$

$P ::= 0 \mid c.P \mid P_1 \mid P_2 \mid n[P] \mid (\nu n)P$

Table 2

$\equiv$ *is the least equivalence relation that* :

i) includes α-conversion
ii) is preserved by all operators except prefixing
iii) satisfies the following rules:

- $P \mid Q \equiv Q \mid P$ (Struct Par Comm)
- $(P \mid Q) \mid R \equiv P \mid (Q \mid R)$ (Struct Par Ass)
- $P \mid 0 \equiv P$ (Struct Zero Par)
- $(\nu n)\ 0 \equiv 0$ (Struct Zero Res)
- $(\nu n)\ (\nu m)\ P \equiv (\nu m)\ (\nu n)\ P$ (Struct Res Res)
- $n \notin fn(P)$ implies $(\nu n)\ (P \mid Q) \equiv P \mid (\nu n)\ Q$ (Struct Res Par)
- $n \neq m$ implies $(\nu n)\ (m[P]) \equiv m[(\nu n)\ P]$ (Struct Res Amb)

$\rightarrow$ *is the least equivalence relation that* :

i) is preserved by all operators except prefixing
ii) satisfies the rules above:

- $m[in\ n.P \mid Q] \mid n[co\text{-}in\ n.R \mid S] \rightarrow n[m[P \mid Q] \mid R \mid S]$ (Red-In)
- $m[n[out\ m.P \mid Q] \mid R] \mid co\text{-}out\ m.S \rightarrow n[P \mid Q] \mid m[R] \mid S$ (Red-Out)
- $open\ n.P \mid n[co\text{-}open\ n.Q \mid R] \rightarrow P \mid Q \mid R$ (Red-Open)
- $P \equiv Q\ ,\ Q \rightarrow R,\ R \equiv S$ implies $P \rightarrow S$ (Red-Struct)

$\rightarrow^*$ is the reflexive and transitive closure of $\rightarrow$.

Table 3

Prefixes1: $\mu ::= in\ n \mid out\ n \mid open\ n \mid co\text{-}in\ n \mid co\text{-}out\ n \mid co\text{-}open\ n \mid pop_m\ n \mid free\ n$

Prefixes2: $\alpha ::= enter_m\ n \mid exit_m\ n \mid co\text{-}enter\ n$

Actions: $\gamma ::= \mu \mid \alpha$

Types: $\sigma ::= \omega \mid \mu.\sigma \mid \alpha.(\nu \tilde{m}) \quad (< \sigma >_n \tau) \mid n[\sigma] \mid (\nu n)\, \sigma \mid \sigma \mid \tau \mid \sigma \wedge \tau$

Table 4

- *Axioms for ω*
  - $\sigma \leq \omega$
  - $\sigma \leq \sigma \mid \omega$
  - $(\nu n)\omega \simeq \omega$
  - $n[\omega] \simeq \omega$

- *Commutativity and Distributivity of parallel composition |*
  - $\sigma \mid \tau \simeq \tau \mid \sigma$
  - $(\sigma \mid \tau) \mid \rho \simeq \sigma \mid (\tau \mid \rho)$

- *Conjunction* $\wedge$
  - $\sigma \wedge \tau \leq \sigma \quad \sigma \wedge \tau \leq \tau$
  - $\sigma \leq \sigma \wedge \sigma$
  - $\sigma \wedge \tau \simeq \tau \wedge \sigma$
  - $\sigma \leq \sigma' \ \& \ \tau \leq \tau' \Rightarrow \sigma \wedge \tau \leq \sigma' \wedge \tau'$
  - $n[\sigma \wedge \tau] \simeq n[\sigma] \wedge n[\tau]$
  - $\rho \mid (\sigma \wedge \tau) \simeq (\rho \mid \sigma) \wedge (\rho \mid \tau)$
  - $\mu.(\sigma \wedge \tau) \simeq \mu.\sigma \wedge \mu\tau$
  - $\alpha.(<\sigma \wedge \tau>_n \rho) \simeq \alpha.(<\sigma>_n \rho) \wedge \alpha.(<\tau>_n \rho)$
  - $\alpha.(<\sigma>_n \rho \wedge \tau) \simeq \alpha.(<\sigma>_n \rho) \wedge \alpha.(<\sigma>_n \tau)$

- *Action*
  - $n[\ in\ n.\ \sigma] \simeq enter\ n.(<m[\ \sigma\ ]>_n \omega)$
  - $m[\ out\ n.\ \sigma] \simeq exit\ n.(<\omega>_n m[\ \sigma\ ])$
  - $n[\ co\text{-}in\ n.\ \sigma] \simeq co\text{-}enter\ n.(<\sigma>_n \omega)$
  - $n[\ co\text{-}open\ n.\ \sigma] \simeq free\ n.\sigma$
  - $n[\ exit\ n.(<\sigma>_n \tau)\ ] \simeq pop\ n.n[\ \sigma\ ] \mid \tau$

- *Reduction*
  - $enter\ n.(\nu\tilde{m})<\sigma_1>_n \sigma_2 \mid co\text{-}enter\ n.(\nu\tilde{q})<\tau_1>_n \tau_2 \leq (\nu\tilde{m})(\nu\tilde{q})(n[\sigma_1 \mid \tau_1] \mid \sigma_2 \mid \tau_2)$
  - $pop\ n.\sigma \mid co\text{-}out\ n.\tau \leq \sigma \mid \tau$
  - $open\ n.\sigma \mid free\ n.\tau \leq \sigma \mid \tau$
- *Restriction*
  - $(\nu m)\ n[\sigma] \simeq n[(\nu m)\ \sigma]$ — $n \neq m$
  - $(\nu m)(\sigma \mid \tau) \simeq \sigma \mid (\nu m)\tau$ — $m \notin fn(\sigma)$
  - $(\nu n)(\nu m)\ \sigma \simeq (\nu m)(\nu n)\ \sigma$
  - $(\nu m)(\sigma \wedge \tau) \simeq (\nu m)\ \sigma \wedge (\nu m)\ \tau$
  - $(\nu m)\ \gamma.\sigma \simeq \omega$ — if $m = fn(\gamma)$
    $\simeq \gamma.(\nu m)\ \sigma$ — if $m \neq fn(\gamma)$

- *Sequentialization*
  - $\mu.\sigma \mid \tau \leq \mu.(\sigma \mid \tau)$
  - $enter\ n.(\nu\tilde{m})(<\sigma>_n \tau) \mid \rho \leq enter\ n.(\nu\tilde{m})(<\sigma>_n (\tau \mid \rho))$
  - $exit\ n.(\nu\tilde{m})(<\sigma>_n \tau) \mid \rho \leq exit\ n.(\nu\tilde{m})(<\sigma \mid \tau>_n \rho)$
  - $co\text{-}enter\ n.(\nu\tilde{m})(<\sigma>_n \tau) \mid \rho \leq co\text{-}enter\ n.(\nu\tilde{m})(<\sigma>_n (\tau \mid \rho))$
  - $\gamma_1.\sigma \mid \gamma_1.\tau \simeq \gamma_1.(\sigma \mid \tau) \wedge \gamma_2.(\sigma \mid \tau)$ — if γ_1 and γ_2 do not match
    $\simeq \sigma' \mid \tau' \wedge \gamma_1.(\sigma \mid \tau) \wedge \gamma_2.(\sigma \mid \tau)$ — if γ_1 and γ_2 match
- *Congruence*
  - $\sigma \leq \tau \Rightarrow n[\sigma] \leq n[\tau]$

- $\sigma \leq \tau \Rightarrow \quad \mu.\sigma \leq \mu.\tau$
- $\sigma \leq \tau \Rightarrow \quad (\nu m)\,\sigma \leq (\nu m)\,\tau$
- $\sigma \leq \tau \Rightarrow \quad \sigma \mid \rho \leq \tau \mid \rho$

- *Transitivity*
  - $\sigma \leq \tau \;\&\; \tau \leq \rho \Rightarrow \quad \sigma \leq \rho$

Table 4 bis

Sequentialization rules

- $\mu \,.\, \sigma \mid \tau \quad \leq \quad \mu.(\sigma \mid \tau)$
- $\mathit{enter}\ \mathsf{n} \,.\, (\nu \tilde{m})\ (< \sigma >_n \tau) \mid \rho \quad \leq \quad \mathit{enter}\ \mathsf{n} \,.\, (\nu \tilde{m})\ (< \sigma >_n (\tau \mid \rho))$
- $\mathit{exit}\ \mathsf{n} \,.\, (\nu \tilde{m})\ (< \sigma >_n \tau) \mid \rho \quad \leq \quad \mathit{exit}\ \mathsf{n} \,.\, (\nu \tilde{m})(< \sigma \mid \tau >_n \rho)$
- $\mathit{co\text{-}enter}\ \mathsf{n} \,.\, (\nu \tilde{m})\ (< \sigma >_n \tau) \mid \rho \quad \leq \quad \mathit{co\text{-}enter}\ \mathsf{n} \,.\, (\nu \tilde{m})\ (< \sigma >_n (\tau \mid \rho))$
- $\mu_1 \,.\sigma \mid \mu_2 \,.\tau \quad \simeq \quad \mu_1.(\sigma \mid \mu_2 \,.\tau) \ \wedge \mu_2.(\mu_1 \,.\sigma \mid \tau)$ if μ_1 and μ_2 do not match

$\simeq$

$\sigma \mid \tau \wedge \ \mu_1.(\sigma \mid \mu_2 \,.\tau) \ \wedge \mu_2.(\mu_1 \,.\sigma \mid \tau) \mu_1.(\sigma \mid \mu_2 \,.\tau) \ \wedge \mu_2.(\mu_1 \,.\sigma \mid \tau)$

if μ_1 and μ_2 match

- $\mu \,.\sigma \mid \ \alpha.(<\rho>_n \tau) \quad \simeq \quad \mu \,.(\ \alpha.(<\rho>_n (\tau \mid \sigma))) \wedge$