

A Calculus of Mobility and Communication for Ubiquitous Computing

Nosheen Gul

Department of Computer Science, University of Leicester, England
ng90@le.ac.uk

We propose a *Calculus of Mobility and Communication (CMC)* for the modelling of mobility, communication and context-awareness in the setting of ubiquitous computing. CMC is an ambient calculus with the *in* and *out* capabilities of Cardelli and Gordon's Mobile Ambients. The calculus has a new form of *global communication* similar to that in Milner's CCS. In CMC an ambient is tagged with a set of ports that agents executing inside the ambient are allowed to communicate on. It also has a new context-awareness feature that allows ambients to query their location. We present reduction semantics and labelled transition system semantics of CMC and prove that the semantics coincide. A new notion of behavioural equivalence is given by defining capability barbed bisimulation and congruence which is proved to coincide with barbed bisimulation congruence. The expressiveness of the calculus is illustrated by two case studies.

1 Introduction

Mark Weiser envisioned [27, 26] that *ubiquitous computing* provides various computing devices available throughout the physical setting. Ubiquitous computing devices are distributed and could be mobile, and interactions among them are concurrent and often depend on the location of the devices. The idea of context-aware computing has originated in [26]. It enables an application to adapt to the changes in its environment and location. Recent advancements in technology have made it possible to detect user's presence or position, or to detect other entities of interest to the user. Therefore, context-awareness and location-awareness have become important features of ubiquitous computing environments.

In literature, a number of formalisms and languages have been introduced for distributed and concurrent systems. Process algebras are used to model formally concurrent systems. Structural Operational Semantics (SOS) is given as a standard approach of defining the semantics of a system by means of transition rules [13, 19]. Several process calculi were developed to model concurrency, communication and distributed systems, most notably CSP [7], CCS [13] and ACP [1]. These process calculi have no primitives to describe certain aspects of behaviours of the ubiquitous computing setting, for example mobility and locations. The idea of mobile code has been formalised by Milner in π -calculus [14]. The aforementioned process calculi do not represent directly physical mobility of devices and their locations or surroundings.

The inspiration for our work comes from several mobile ambient and process calculi that have proved useful in the modelling of mobility, communication and structure of systems. The calculus of *Mobile Ambients*, *MA* for short, [4] is a process calculus for modelling mobile agents over wide-area networks. In *MA* the ambients represent mobile, nested, computational structures with local communication. Ambients are named terms of the form $n[P]$ where n is a name and P a process.

In smart indoor settings, spatial organisation is considered an important object for providing communication among various fixed and mobile structures. Despite the advances in the ubiquitous and mobile

computing, it is fundamental to formally model physical mobility of devices and interactions among mobile agents that may communicate globally. Communication in such settings could be global, which means that agents may interact with subagents inside other agents. Moreover, the structures in such settings may be mobile, and may need to have knowledge of their current location and surroundings. In order to model such attributes of ubiquitous computing, this paper presents a *Calculus of Mobility and Communication (CMC)*. In CMC, mobility, global communication and location-awareness are considered as first class entities. According to [4], MA was proposed to model mobility and locations that could not be modelled directly by other traditional calculi like Milner’s *Calculus of Communicating Systems (CCS)*, therefore we model locations and mobility as in Mobile Ambients. In MA, ambients may enter or exit named ambients by their *in n* and *out n* capabilities. The ambient’s *open* capability dissolves its boundary so that the communication may take place locally. We do not use the *open* capability in this paper since we introduce a new mechanism of global communication. CMC aims at adding global communication as in Milner’s CCS. To achieve this we define ambients as $m_A[P]$, where m is the name of the ambient, A is the set of ports that m is allowed to communicate on, and P is an executing agent. This helps in modelling the globally communicating mobile agents in the setting of ubiquitous computing.

We develop Structural Operational Semantics for ambients mobility and reuse the CCS rules with an additional rule to introduce global communication. As in MA, we also show ambients mobility by means of a reduction semantics. For global communication, in contrast, it would be a challenge to find simple and intuitive reduction rules since (global) communication can happen via an arbitrary number of ambients that could be located far-away in the structure of a term, and it depends on whether or not all these ambients allow the communication. This is however unsound and could derive reductions matching no corresponding transitions. We also develop a new notion of behavioural equivalence for our calculus, and formulate the equivalence in terms of α -transitions and observation predicate, inspired by [4, 10]. Thus, we define barbed bisimulation and congruence, and capability barbed bisimulation and congruence, we then prove that the respective congruence relations of the two forms of barbs imply each other. The work on behavioural equivalence is still in progress, and the recent advances in the behavioural semantics theory of mobile ambients, as in [10, 11], should be useful.

Context-awareness is an essential paradigm of ubiquitous computing environment that makes applications adaptive with their surroundings, and enables processes to be aware of the setting in which they are being executed. Therefore, we further extend the syntax of the calculus and add a context-awareness mechanism by introducing two capabilities to it. The new capability $ploc(x).P$ allows an ambient to acquire the name of its parent, whereas $sloc(x).P$ enquires the sibling’s name of an ambient. This feature empowers ambients to have knowledge of their current location and surroundings.

The rest of the paper is organised as follows: We introduce our basic calculus, CMC_b , in Section 2 where its reduction semantics and labelled transition semantics are given. We show that the two types of semantics coincide for an appropriate sub-calculus of CMC_b . In the same section we define behavioural equivalence for the calculus. Section 3 presents intelligent hospital case study to illustrate the usefulness of CMC_b . Section 4 extends CMC_b to CMC with the $ploc(x)$ and $sloc(x)$ capabilities. We present operational semantics for the new capabilities and argue that the semantics coincide. In Section 5 we give a shopping mall case study to illustrate the usefulness of CMC. Section 6 contains conclusions.

2 A Calculus of Mobility and Communication

We introduce the syntax of the basic part of CMC, denoted by CMC_b , in Tables 1 and 2. Informally, CMC_b inherits its syntax from MA and CCS. The syntax allows global communication among ambients

<i>Names :</i>	$m_A, n_B, k_C \dots \in \mathcal{N}$				
<i>Actions :</i>	$\alpha, \beta, \dots \in Act = \mathcal{A} \cup \overline{\mathcal{A}} \cup \{\tau\}$				
<i>Variables :</i>	$x, y, \dots \in \mathcal{X}$				
<i>Processes :</i>	$P, Q ::=$	D	$C.P$	$a(z).P$	$\bar{a}(x).P$
		$ m_A[P]$	$ P+Q$	$ P Q$	$ (vm_A)P$
		$ (\nu l)P$	$ P[f]$		
<i>Capabilities :</i>	$C ::=$	x	μ	ε	$C.C'$

Table 1: Syntax of CMC_b

that could be mobile. We assume that \mathcal{A} is an infinite set of port names, which is ranged over by a, b, c , and the set of co-names, denoted by $\overline{\mathcal{A}}$ is ranged over by $\bar{a}, \bar{b}, \bar{c}$. We set $\mathcal{L} = \mathcal{A} \cup \overline{\mathcal{A}}$ and let A, B, C range over it. An infinite set Act comprises all possible actions that an agent can perform and α, β range over it. Act also includes τ , which is a single completed action of composite agents. So $Act = \mathcal{L} \cup \{\tau\}$, and the typical subsets of α are A, B . The set of agent constants \mathcal{K} is ranged over by D and E , and the deadlocked agent 0 is a member of \mathcal{K} . In our syntax the variable z in $a(z).P$ can be replaced by a value from a set \mathcal{V} , which may contain the capabilities as defined in Table 1.

For the mobility part of CMC_b syntax in Table 1, we assume an infinite set of ambient names \mathcal{N} that is ranged over by m_A, n_B and k_C , where $A, B, C \subseteq \mathcal{A} \cup \overline{\mathcal{A}}$. We define our ambient as a term $m_A[P]$, where, m is the name of the ambient, A is the set of ports that ambient m is allowed to communicate on, and P is an executing agent. When ambients allow communication on all visible ports then we shall write $m[P]$ instead of $m_A[P]$. Other ambient constructs that are inherited from MA are $(vm)P$, $C.P$ and $C.C'$. An ambient restriction $(vm)P$ executes process P with a private ambient named m . In $C.P$, the process P cannot start execution until the prefix capability C is performed. The capability μ in Table 2 allows ambients to perform certain actions, namely *in* and *out*, whereas $C.C'$ represents a sequence of capabilities (path) when input variable represents one or more of these capabilities. The empty path is represented by ε .

We further borrow the constructs for agent constants, action prefixing, parallel composition, summation and action restriction from Milner's CCS or the π -calculus [13, 14]. The agent constant D has a unique equation of the form $D \stackrel{def}{=} P$ where P is an agent that may contain agent constants. The agent constants can also be defined in terms of each other. $\bar{a}(x)$ and $a(z).P$ sends or receives a message on port \bar{a} and a respectively, and then execute P . The received message can be any value $v \in V$, and is bound to the variable z in P . Parallel composition is given in terms of a binary operator, $P|Q$, and summation is given by the choice operator $P+Q$ that allows either process P or process Q to execute. In $(\nu l)P$ the port labels l or \bar{l} are restricted in P , where $l \in \mathcal{L}$. In a relabelling $P[f]$, P is a process with the relabelling function f applied to its action labels. Finally, we have the set of terms $T(\Sigma, V)$, where V is the set of process variables, and $T(\Sigma)$, the set of closed terms (agents or processes) ranged over by P, Q .

2.1 Reduction Semantics of CMC_b

The reduction semantics is formalised by two concepts: the structural congruence relation, \equiv , and the reduction relation \rightarrow . We follow the definition in [10].

Definition 1. A relation \mathcal{R} over processes in a process calculus is contextual, if it is preserved by all the operators in the process calculus. A relation \mathcal{R} over processes in a process calculus is p-contextual w.r.t a set of operators Op , if it is preserved by all the operators in the set Op .

<i>Ambient Prefixes</i> :	$\mu ::= in_{n_B}$		out_{n_B}		
<i>Action Prefixes</i> :	$\alpha ::= a(z)$		$\bar{b}(z)$		τ
<i>Ambient Actions</i> :	$\lambda ::= enter_{n_B}$		$move_{n_B}$		$exit_{n_B}$ μ
<i>Labels</i> :	$\ell ::= \mu$		α		λ τ
<i>Outcomes</i> :	$O ::= P$		K		
<i>Concretions</i> :	$K ::= (v\tilde{m})\langle P \rangle Q$				

Table 2: Prefixes and labels

We denote the set of all names occurring free in P by $fn(P)$.

Definition 2. Structural congruence, \equiv , over CMC_b processes is the least p-contextual equivalence relation w.r.t the set of operators $Op_1 = \{v, |, [f], n_B[\], C., \alpha.\}$, where C and α are in Tables 1 and 2, that satisfies the following axioms:

$P Q \equiv Q P$	(ParComm)	$A \equiv P$ if $A \stackrel{def}{=} P$	(Const)
$(P Q) R \equiv P (Q R)$	(ParAssoc)	$(vn_B)(P Q) \equiv P (vn_B)Q$ if $n_B \notin fn(P)$	(ResPar)
$P 0 \equiv P$	(ZeroPar)	$(vn_B)(m_A[P]) \equiv m_A[(vn_B)P]$ if $n \neq m$	(ResAmb)
$P + Q \equiv Q + P$	(SumComm)	$(vn_B)(vm_A)P \equiv (vm_A)(vn_B)P$	(ResRes)
$(P + Q) + R \equiv P + (Q + R)$	(SumAssoc)	$(vn_B)0 \equiv 0$	(ZeroRes)
$P + 0 \equiv P$	(ZeroIdentity)	$\varepsilon.P \equiv P$	(Epsilon)

Definition 3. The reduction relation, \rightarrow , over CMC_b processes is the least p-contextual relation w.r.t the set of operators $Op_2 = \{v, |, n_B[\]\}$ that satisfies the rule and axioms in Table 3.

$m_A[in_{n_B}.P Q] n_B[R] \rightarrow n_B[m_A[P Q] R]$	(Red In)
$n_B[m_A[out_{n_B}.P Q] R] \rightarrow m_A[P Q] n_B[R]$	(Red Out)
$P \equiv Q, Q \rightarrow Q', Q' \equiv P' \Rightarrow P \rightarrow P'$	(Red \equiv)

Table 3: Reduction axioms and rule for CMC_b

The axiom Red In in Table 3 shows how an ambient m_A may enter into an ambient n_B by the virtue of its in_{n_B} capability. The reduction transforms m_A , which is a sibling ambient of n_B , into a child of n_B . The axiom Red Out describes emigration of an ambient m_A from an ambient n_B by performing the out_{n_B} capability. The reduction transforms m_A , which is a child of n_B , to a sibling of n_B .

2.2 Labelled Transition System for CMC_b

A *labelled transition system (LTS)* is a tuple $(S, L, \{\xrightarrow{l} : l \in L\})$, where S is a set of states, L is a set of transition labels, and \xrightarrow{l} are transition relations, one for each $l \in L$. The LTS for CMC_b is given as follows: The set of processes of CMC_b is the set of states, the set of labels α as in Table 2 is the set of transition labels, and the transition relations $\xrightarrow{\alpha}$ are defined by Plotkin's *Structural Operational Semantics (SOS)* [19] rules in Tables 4 and 5. In our semantics $P \xrightarrow{\tau} Q$ represents not only binary communication of processes as in CCS but also mobility of ambients by means of their in_{n_B} and out_{n_B} capabilities. In order to model mobility by τ -transitions additional labels and auxiliary terms are used,

namely labels λ and concretions K in Table 2. So we will need auxiliary transitions $P \xrightarrow{\lambda} O$, where P is a process, λ is a label and O represents an outcome in Table 2, which is either a process or *concretion* of the form $(v\tilde{m})\langle P \rangle Q$ as introduced by Milner [14] and used by Merro and Hennessy [10]. We adopt the following convention after [10]. If K is the concretion $v\tilde{m}\langle P \rangle Q$, then vuK stands for $v(u\tilde{m})\langle P \rangle Q$, if $u \in fn(P)$, otherwise $v\tilde{m}\langle P \rangle vu(Q)$. A similar convention is followed for λ -Par in Table 5. We define $K | P'$ as the concretion $v\tilde{m}\langle P \rangle (Q | P')$ where using α -conversion if necessary, \tilde{m} is selected in such a way that $fn(P') \cap \tilde{m} = \emptyset$.

$$\begin{array}{c}
\text{(Act)} \frac{}{\mu.P \xrightarrow{\mu} P} \qquad \text{(Enter)} \frac{P \xrightarrow{in n_B} P'}{m_A[P] \xrightarrow{enter n_B} \langle m_A[P'] \rangle 0} \qquad \text{(Co-Enter)} \frac{}{n_B[P] \xrightarrow{move n_B} \langle P \rangle 0} \\
\text{(\tau-In)} \frac{P \xrightarrow{enter n_B} (v\tilde{p})\langle P' \rangle P'' \quad Q \xrightarrow{move n_B} (v\tilde{q})\langle Q' \rangle Q'' \quad (*)}{P | Q \xrightarrow{\tau} (v\tilde{p})(v\tilde{q})(n_B[P' | Q'] | P'' | Q'')} \\
\text{(Exit)} \frac{P \xrightarrow{out n_B} P'}{m_A[P] \xrightarrow{exit n_B} \langle m_A[P'] \rangle 0} \qquad \text{(\tau-Out)} \frac{P \xrightarrow{exit n_B} (v\tilde{m})\langle P' \rangle P'' \quad (**)}{n_B[P] \xrightarrow{\tau} (v\tilde{m})(P' | n_B[P''])}
\end{array}$$

Table 4: Transition rules for mobility. Conditions (*) and (**) are defined as follows:

(*) $(fn(P') \cup fn(P'')) \cap \tilde{q} = (fn(Q') \cup fn(Q'')) \cap \tilde{p} = \emptyset$ and (**) $(fn(P') \cup fn(P'')) \cap \tilde{m} = \emptyset$

$$\begin{array}{c}
\text{(\lambda-Par)} \frac{P \xrightarrow{\lambda} O}{P | Q \xrightarrow{\lambda} O | Q} \quad (*) \qquad \text{(\lambda-Res)} \frac{P \xrightarrow{\lambda} O}{(vu)P \xrightarrow{\lambda} (vu)O} \quad (u \notin fn(\lambda)) \quad (*) \\
\text{(\tau-Amb)} \frac{P \xrightarrow{\tau} P'}{n_A[P] \xrightarrow{\tau} n_A[P']} \qquad \text{(Struct)} \frac{P \equiv Q \quad Q \xrightarrow{l} Q' \quad Q' \equiv P'}{P \xrightarrow{l} P'}
\end{array}$$

Table 5: Transition rules for other operators of CMC_b . Condition (*) says that the definition of λ is extended to include also a τ .

Transitions $P \xrightarrow{\lambda} O$ are not first class transitions; they are only helpful in SOS rules that define τ -transitions of processes corresponding to the movement by *in* n_B and *out* n_B capabilities.

Communication in CMC_b is defined as in CCS, so in addition to the SOS rules in Tables 4 and 5, we have the SOS rules for CCS as in [13] (also in Appendix A) and the following Global-Com rule:

$$\text{(Global-Com)} \frac{P \xrightarrow{\alpha} P'}{m_A[P] \xrightarrow{\alpha} m_A[P']} \quad (\alpha \neq \tau \text{ and if } (\alpha = a(x) \text{ or } \alpha = \bar{a}(x) \text{ for some } a) \text{ then } a \in A)$$

Global-Com allows ambients to communicate globally only on ports $a \in A$. Recall that when ambients allow communication on all visible channels then we shall write $m[P]$ instead of $m_A[P]$.

Next, we discuss some reductions and at the same time explain how auxiliary labels and transitions are used in defining mobility transitions. We assume $m_A[in n_B.P] | Q | n_B[R]$ for some P, Q and R .

The ambient m_A , for some A , has the capability to enter an ambient n_B for some B . By Red In axiom in Table 3 we have,

$$m_A[in n_B.P] | Q | n_B[R] \longrightarrow n_B[m_A[P] | R] | Q.$$

We now derive the τ -transition of $m_A[in\ n_B.P] \mid Q \mid n_B[R]$ by τ -In rule in Table 4. For simplicity, we assume that there are no private names in Q and R . We have $in\ n_B.P \xrightarrow{in\ n_B} P$. When the migration occurs, we must identify the moving ambient m_A , and the agent that is left behind. To model these two agents we use concretion $\nu\tilde{m}\langle P \rangle Q$, where P is the agent that moves, while Q is the agent that stays behind, and \tilde{m} is the set of private names shared by P and Q . We introduce a new action $enter\ n_B$ and have $m_A[in\ n_B.P] \xrightarrow{enter\ n_B} \langle m_A[P] \rangle 0$. By λ -Par in Table 5 we obtain $m_A[in\ n_B.P] \mid Q \xrightarrow{enter\ n_B} \langle m_A[P] \rangle (0 \mid Q) \equiv \langle m_A[P] \rangle Q$.

Next, to achieve the τ -transition there must exist a sibling ambient n_B . We define a new action $move\ n_B$ for n_B to complete this interaction. By τ -In we get,

$$m_A[in\ n_B.P] \mid Q \mid n_B[R] \xrightarrow{\tau} n_B[m_A[P] \mid R] \mid Q.$$

After the transition the ambient m_A , becomes a child of n_B .

Next, we explain emigration capability by considering $m_A[n_B[out\ m_A.P] \mid Q]$, for some P and Q where Q has no private names. The ambient n_B may emigrate from m_A by its $out\ m_A$ capability. By Red Out we have,

$$m_A[n_B[out\ m_A.P] \mid Q] \rightarrow n_B[P] \mid m_A[Q].$$

We derive the τ -transition of $m_A[n_B[out\ m_A.P] \mid Q]$ by τ -Out. We define a new action $exit\ m_A$, and by Exit in Table 4 we get $n_B[out\ m_A.P] \xrightarrow{exit\ m_A} \langle n_B[P] \rangle 0$. By λ -Par we get $n_B[out\ m_A.P] \mid Q \xrightarrow{exit\ m_A} \langle n_B[P] \rangle Q$, which shows that when this capability is exercised $n_B[P]$ moves out, while the process Q remains inside m_A . By τ -Out we have,

$$m_A[n_B[out\ m_A.P] \mid Q] \xrightarrow{\tau} n_B[P] \mid m_A[Q].$$

After the transition the ambient n_B , becomes a sibling of m_A .

2.3 Results

In this subsection we show that the LTS semantics (SOS semantics) coincides with the reduction semantics for a sub-calculus T' of CMC_b that consists of all operators of CMC_b apart from the prefixing with actions (including τ) operators, the choice operator and the relabelling operator.

Since we have developed operational semantics for the mobility part of CMC , therefore we intuitively restrict equivalence between the operational semantics to a subset of the calculus, and prove the soundness and completeness of the semantics. Soundness ensures that for every reduction of a T' term there is a valid τ -transition of the term, and the target of the τ -transition is congruent to the target of the reduction. Completeness ensures that for every valid τ -transition of a T' term there is a valid reduction of the term, and the targets of the τ -transitions and the reductions are the same.

Theorem 1. (a) $\forall P, P' \in T'. P \rightarrow P' \implies \exists Q \in T'. P \xrightarrow{\tau} Q \equiv P'$. (b) $\forall P, R \in T'. P \xrightarrow{\tau} R \implies P \rightarrow R$.

Proof. By transition induction where we consider cases of reductions or transitions of terms depending on the structure of the terms. The proof of part (a) is straightforward, whereas to show part (b) a number of auxiliary lemmas are required, similarly as in [10]. Given a transition $P \xrightarrow{\lambda} O$, where λ is as in Table 2, these lemmas state the structure of terms P and O . For example, for ambient entering capability we require the following lemmas:

Lemma 1. If $P \xrightarrow{enter\ n_B} \nu\tilde{p}\langle P' \rangle P''$ then $P \equiv \nu\tilde{p}(k_A[in\ n_B.P_1 \mid P_2] \mid P_3)$, $P' \equiv k_A[P_1 \mid P_2]$ and $P'' \equiv P_3$, for some P_1, P_2, P_3, k_A with $n_B \notin \tilde{p}$, where \tilde{p} is a set of private ambient names in P .

Lemma 2. If $Q \xrightarrow{move\ n_B} v\tilde{q} \langle Q' \rangle Q''$ then $Q \equiv v\tilde{q} (n_B[Q_1] \mid Q_2)$, $Q' \equiv Q_1$ and $Q'' \equiv Q_2$, for some Q_1, Q_2 , with $n_B \notin \tilde{q}$, where \tilde{q} is a set of private ambient names in Q .

The detailed proof of Theorem 1 is given in [6] (Section 4.2). \square

2.4 Behavioural Semantics

We develop an appropriate notion of behavioural equivalence for CMC_b . All processes and context mentioned in this section are from our calculus CMC_b . We formulate the equivalence in terms of α -transitions ($\xrightarrow{\alpha}$), for $\alpha \in a(z), \bar{b}(z), in\ m_A, out\ m_A, \tau$, for all a, b, m, A , and observation predicate as in [4, 10]. We write $P \downarrow_{n_A}$ to denote the presence of ambient n_A at the top level, in the other words process P may interact with the environment via n_A . We write $P \Downarrow_{n_A}$, if after some number of τ -transitions, the process P exhibits n_A at the top level.

Definition 4. (Barbs)

$P \downarrow_{n_A} \stackrel{def}{=} P \equiv v\tilde{m}(n_A[P_1] \mid P_2)$, where $n_A \notin \tilde{m}$ for some P_1, P_2 and
 $P \Downarrow_{n_A} \stackrel{def}{=} P \xrightarrow{\hat{\tau}} Q$ and $Q \downarrow_{n_A}$ for some Q .

Definition 5. (Barb Preserving)

A relation R over processes is said to be *barb preserving* if $P R Q$ and $P \downarrow_{n_A}$ implies $Q \Downarrow_{n_A}$.

Definition 6. (Context)

A *context* $\mathcal{C}[\cdot]$ is a process with zero or more holes $[\cdot]$. A hole $[\cdot]$ in a context \mathcal{C} is replaced by at most one occurrence of a process. A context $\mathcal{C}[\cdot]$ with a hole $[\cdot]$ replaced by a process P is denoted by $\mathcal{C}[P]$.

Definition 7. (Contextual Equivalence)

Processes P, Q are *contextual equivalent*, denoted by $P \simeq Q$, if for all contexts $\mathcal{C}[\cdot]$ and ambient names n_A , $\mathcal{C}[P] \downarrow_{n_A}$ implies $\mathcal{C}[Q] \downarrow_{n_A}$.

Since we are considering weak equivalence, we provide the notion of weak actions as follows. We write $\alpha \in Act$ (recall that $\tau \in Act$). We write \Rightarrow for the reflexive and transitive closure of $\xrightarrow{\tau}$, where $\xrightarrow{\tau}$ specifies exactly the τ -transition. $\xrightarrow{\hat{\tau}}$ specifies at least a τ transition. $\hat{\alpha}$ is a sequence obtained by deleting all occurrences of τ actions, note that $\hat{\tau} = \varepsilon$. Furthermore, $\xrightarrow{\hat{\tau}}$ is $\xrightarrow{\varepsilon}$, an empty sequence of τ -transitions, and $\xrightarrow{\hat{\alpha}}$ is $\xrightarrow{\alpha}$, for $\alpha \neq \tau$.

We define two forms of barbs; one at ambient level whereas another for ambients capabilities. They give rise to (a) barbed bisimulation and congruence, and (b) capability barbed bisimulation and congruence. We then show that the respective congruence relations imply each other.

Definition 8. (Barbed Bisimulation and Congruence)

A relation S is a *barbed bisimulation*, if it is symmetric and if $(P, Q) \in S$ then for all $\alpha \in \{a(z), \bar{b}(z), in\ m_A, out\ m_A\}$,

- if $P \xrightarrow{\alpha} P'$ then $Q \xrightarrow{\hat{\alpha}} Q'$ and $(P', Q') \in S$;
- if $P \downarrow_{n_A}$ then $Q \Downarrow_{n_A}$.

Processes P and Q are barbed bisimilar, $P \approx Q$, if $(P, Q) \in S$ for some barbed bisimulation S . P and Q are barbed congruent, $P \cong Q$, if for all contexts $\mathcal{C}[\cdot]$, $\mathcal{C}[P] \approx \mathcal{C}[Q]$.

Definition 9. We write $P \downarrow_{\beta}$ if $P \xrightarrow{\beta} P'$ for some P' , where $\beta \in \{in\ n_A, out\ n_A, enter\ n_A, move\ n_A, exit\ n_A\}$.

We write $P \Downarrow_{\beta}$ if $P \xrightarrow{\tau^*} P' \xrightarrow{\beta} P''$ for some P' and P'' .

Definition 10. (Capability Barbed Bisimulation)

Let $L = \{in\ n_A, out\ n_A, enter\ n_A, move\ n_A, exit\ n_A\}$, and let $\beta \in L$. A relation R is a β -barbed bisimulation, if R is symmetric and if $(P, Q) \in R$ then for all $\alpha \in \{a(z), \bar{b}(z), in\ n_A, out\ n_A\}$:

- if $P \xrightarrow{\alpha} P'$ then $Q \xrightarrow{\hat{\alpha}} Q'$ and $(P', Q') \in R$;
- if $P \Downarrow_{\beta}$ then $Q \Downarrow_{\beta}$.

P and Q are β -barbed bisimilar, $P \approx_{\beta} Q$, if $(P, Q) \in R$ for some β -barbed bisimulation R . P and Q are barbed congruent, $P \cong_{\beta} Q$, if for all contexts $\mathcal{C}[\cdot]$, $\mathcal{C}[P] \approx_{\beta} \mathcal{C}[Q]$.

We now prove that two congruence relations, namely barbed bisimulation congruence and capability barbed bisimulation congruence imply each other.

Theorem 2. Let $P, Q \in \text{CMC}_b$. Then, $P \cong Q$ iff $P \cong_{\beta} Q$ for all n_B .

Proof. We consider a case where $\beta = move\ n_B$, and show that $P \cong Q$ implies $P \cong_{move\ n_B} Q$ for all P, Q and n_B .

Assume that $P \cong Q$ and $P \Downarrow_{move\ n_B}$, and we will show $Q \Downarrow_{move\ n_B}$. We define a context $\mathcal{C}_1[\cdot]$ as follows:

$$\mathcal{C}_1[\cdot] \stackrel{def}{=} \nu m_A([\cdot]) \mid \nu a(k_C[in\ n_B.out\ n_B.\bar{a}.0] \mid a.m_A[P]), \text{ with } a \notin B \text{ and } a \in C$$

Global communication is very useful in the definition of context $\mathcal{C}_1[\cdot]$. It acts as a guard and the context may interact with the environment via corresponding guarded ambient if the guard is satisfied.

Before we continue with proof of Theorem 2, we shall need the following lemma.

Lemma 3. For m_A and k_C fresh in an agent R , $R \Downarrow_{move\ n_B}$ iff $\mathcal{C}_1[R] \Downarrow_{m_A}$.

Proof. We show, $R \Downarrow_{move\ n_B}$ implies $\mathcal{C}_1[R] \Downarrow_{m_A}$

By Definition 9, $R \Downarrow_{move\ n_B}$ implies $R \xrightarrow{\tau^*} R' \xrightarrow{move\ n_B} R''$ for some R', R'' . Since $R \Downarrow_{move\ n_B}$ is valid, we obtain $R \xrightarrow{\tau^*} R' \xrightarrow{move\ n_B} R''$.

We consider $R' \xrightarrow{move\ n_B} R''$. By Lemma 2, if $R' \xrightarrow{move\ n_B} \nu \tilde{r}\langle Q' \rangle Q''$ then $R' \equiv \nu \tilde{r}(n_B[R_1] \mid R_2)$ and $R'' \equiv \nu \tilde{r}\langle Q' \rangle Q''$, where $Q' \equiv R_1$ and $Q'' \equiv R_2$. We now have,

$$\mathcal{C}_1[R'] \equiv \mathcal{C}_1[\nu \tilde{r}(n_B[R_1] \mid R_2)] \equiv \nu m_A(\nu \tilde{r}(n_B[R_1] \mid R_2)) \mid \nu a(k_C[in\ n_B.out\ n_B.\bar{a}.0] \mid a.m_A[P])$$

Since by (*) in τ -In in Table 4, the members of \tilde{r} are not free names in $\nu a(k_C[in\ n_B.out\ n_B.\bar{a}.0] \mid a.m_A[P])$, and $a \notin fn(\nu m_A(\nu \tilde{r}(n_B[R_1] \mid R_2)))$, the process $\mathcal{C}_1[\nu \tilde{r}(n_B[R_1] \mid R_2)]$ executes as follows

$$\begin{aligned} &\xrightarrow{\tau} \nu a \nu \tilde{r}(\nu m_A(n_B[k_C[out\ n_B.\bar{a}.0] \mid R_1] \mid R_2) \mid a.m_A[P]), \\ &\quad (k_C \neq m_A \text{ and } k_C \notin \tilde{r}) \text{ and } (a \notin fn(R_2) \text{ and } \tilde{r} \cap fn(P) = \emptyset) \quad (\tau\text{-In}) \\ &\xrightarrow{\tau} \nu a \nu \tilde{r}(\nu m_A(n_B[R_1] \mid R_2 \mid k_C[\bar{a}.0]) \mid a.m_A[P]) \quad (\tau\text{-Out}) \\ &\xrightarrow{\tau} \nu a \nu \tilde{r}(\nu m_A(n_B[R_1] \mid R_2 \mid k_C[0]) \mid m_A[P]) \quad (\text{Global-Com}) \end{aligned}$$

We need to show $\mathcal{C}_1[R] \Downarrow_{m_A}$ which by our predicate definition, means $\mathcal{C}_1[R] \xrightarrow{\tau^*} \mathcal{C}_1[R'] \Downarrow_{m_A}$, and $\mathcal{C}_1[R'] \Downarrow_{m_A}$ means $\mathcal{C}_1[R'] \equiv \nu \tilde{m}(m_A[P_1] \mid P_2)$ for some P_1, P_2, \tilde{m} . When $P_2 \equiv \nu m_A(n_B[R_1] \mid R_2 \mid k_C[0])$, $m_A[P_1] \equiv m_A[P]$ and $\tilde{m} \equiv \nu a \nu \tilde{r}$, then we obtain $\mathcal{C}_1[R'] \equiv \nu a \nu \tilde{r}(\nu m_A(n_B[R_1] \mid R_2 \mid k_C[0]) \mid m_A[P])$, which implies $\mathcal{C}_1[R'] \Downarrow_{m_A}$. Since $R \xrightarrow{\tau^*} R'$ we obtain $\mathcal{C}_1[R] \xrightarrow{\tau^*} \mathcal{C}_1[R']$. Since $\mathcal{C}_1[R] \xrightarrow{\tau^*} \mathcal{C}_1[R']$ and $\mathcal{C}_1[R'] \Downarrow_{m_A}$, we obtain $\mathcal{C}_1[R] \Downarrow_{m_A}$ as required.

We now show, $\mathcal{C}_1[R] \Downarrow_{m_A}$ implies $R \Downarrow_{move\ n_B}$

Since $\mathcal{C}_1[R] \Downarrow_{m_A}$ means $\mathcal{C}_1[R] \xrightarrow{\tau^*} \mathcal{C}_1[R'] \Downarrow_{m_A}$ for some R' , we have

$$\mathcal{C}_1[R] \equiv \nu m_A(R) \mid \nu a(k_C[in\ n_B.out\ n_B.\bar{a}.0] \mid a.m_A[P])$$

Here, $\mathcal{C}_1[R]$ may interact with the environment via the ambient m_A only if, after some τ -transitions, m_A exists at the top level. Therefore, $R \xrightarrow{\tau^*} R' \Downarrow_{n_B}$ and we obtain

$$\nu m_A(R') \mid \nu a(k_C[in\ n_B.out\ n_B.\bar{a}.0] \mid a.m_A[P])$$

Since we define predicate $(R' \Downarrow_{n_B})$ as $R' \Downarrow_{n_B} \stackrel{def}{=} R' \equiv \nu \tilde{q}(n_B[Q_1] \mid Q_2)$ for some Q_1, Q_2 , and $n_B \notin \tilde{q}$, we obtain

$$\nu m_A(R') \mid \nu a(k_C[in\ n_B.out\ n_B.\bar{a}.0] \mid a.m_A[P]) \xrightarrow{\tau^*} \nu m_A(R') \mid \nu a(k_C[0] \mid m_A[P]).$$

Since after a number of τ -transitions we have m_A at the top level of context \mathcal{C}_1 , so $\mathcal{C}_1[R']$ may interact with environment via m_A and we obtain $\mathcal{C}_1[R'] \Downarrow_{m_A}$. Since $\mathcal{C}_1[R] \xrightarrow{\tau^*} \mathcal{C}_1[R']$ and $\mathcal{C}_1[R'] \Downarrow_{m_A}$, we obtain $\mathcal{C}_1[R] \Downarrow_{m_A}$.

Since we have $R' \equiv \nu \tilde{q}(n_B[Q_1] \mid Q_2)$, we show $\xrightarrow{move\ n_B}$ as follows:

$$\begin{array}{c} \text{Co-Enter} \frac{}{n_B[Q_1] \xrightarrow{move\ n_B} \langle n_B[Q_1] \rangle 0} \\ \lambda\text{-Par} \frac{}{n_B[Q_1] \mid Q_2 \xrightarrow{move\ n_B} \langle n_B[Q_1] \rangle (0 \mid Q_2)} \\ \lambda\text{-Res} \frac{}{\nu \tilde{q}(n_B[Q_1] \mid Q_2) \xrightarrow{move\ n_B} \nu \tilde{q}\langle n_B[Q_1] \rangle (0 \mid Q_2) \equiv \nu \tilde{q}\langle n_B[Q_1] \rangle Q_2} \\ \text{Struct} \frac{}{\nu \tilde{q}(n_B[Q_1] \mid Q_2) \xrightarrow{move\ n_B} \nu \tilde{q}\langle n_B[Q_1] \rangle Q_2} \end{array}$$

□

Now we return to the proof of Theorem 2. Since $P \Downarrow_{move\ n_B}$ we get, by Lemma 3, $\mathcal{C}_1[P] \Downarrow_{m_A}$. Since $P \cong Q$, we obtain $\mathcal{C}_1[P] \cong \mathcal{C}_1[Q]$, for context $\mathcal{C}_1[\cdot]$. Then since $\mathcal{C}_1[P] \cong \mathcal{C}_1[Q]$, $\mathcal{C}_1[P] \Downarrow_{m_A}$ gives us $\mathcal{C}_1[Q] \Downarrow_{m_A}$. Finally, by Lemma 3, $\mathcal{C}_1[Q] \Downarrow_{m_A}$ implies $Q \Downarrow_{move\ n_B}$ as required.

Next, we show the right to left implication, namely $P \cong_{move\ n_B} Q \Rightarrow P \cong Q$ for all P, Q . Assume that $P \cong_{move\ n_B} Q$ and $P \Downarrow_{m_A}$, and we will show $Q \Downarrow_{m_A}$. We define the context $\mathcal{C}_2[\cdot]$ as follows:

$$\mathcal{C}_2[\cdot] \stackrel{def}{=} \nu n_B([\cdot] \mid \nu a(k_C[in\ m_A.out\ m_A.\bar{a}.0] \mid a.n_B[P])), \text{ with } a \notin A \text{ and } a \in C.$$

Lemma 4. For k_C and n_B fresh in an agent R , $R \Downarrow_{m_A}$ iff $\mathcal{C}_1[R] \Downarrow_{move\ n_B}$.

Proof. Since the proof is very similar to the proof of Lemma 3 it is omitted. □

Since $P \Downarrow_{m_A}$ Lemma 4 gives us $\mathcal{C}_2[P] \Downarrow_{move\ n_B}$. Since $P \cong_{move\ n_B} Q$, we obtain $\mathcal{C}_2[P] \cong_{move\ n_B} \mathcal{C}_2[Q]$ for context $\mathcal{C}_2[\cdot]$. Next, since $\mathcal{C}_2[P] \cong_{move\ n_B} \mathcal{C}_2[Q]$, $\mathcal{C}_2[P] \Downarrow_{move\ n_B}$ gives us $\mathcal{C}_2[Q] \Downarrow_{move\ n_B}$. Hence, by Lemma 4, $\mathcal{C}_2[Q] \Downarrow_{move\ n_B}$ implies $Q \Downarrow_{m_A}$ as required. □

Conjecture 1. We conjecture that Theorem 2 will hold for the other capabilities, namely *enter* n_B and *exit* n_B of CMC_b.

3 Intelligent Hospital Case Study

This case study illustrates the usefulness of CMC_b in the given problem domain. Agents' mobility and global communication features are modelled in a scenario where services follow mobile ambients, and server supplies services globally to appropriate device provided that the receiving ambient is at the same location as the device.

We consider a hospital building and a doctor who moves around the building and helps patients. While dealing with patients, he may need to use information displayed on screens that are fixed around the building. We assume that an independent server communicates globally with the doctor and with the screens around the building. The purpose of this case study is to model services following the doctor around the building, more specifically to ensure that information is shown only on the screens of the rooms where the doctor is present.

An ambient k represents the building. The ambient k contains ambients dr_K and w_L which represent the doctor's room and the ward respectively. K and L are sets of communication ports, where $b, c_1 \in K$ and $b, c_2 \in L$. This means that the ambient dr_K can communicate at least on ports a and c_1 and the ambient w_L can communicate at least on ports a and c_2 . Furthermore, there are two fixed screens scr_{A_1} and scr_{A_2} in dr_K and w_L respectively. A_1 and A_2 are the sets of communication ports, where $c_1 \in A_1$ and $c_2 \in A_2$, but $c_1 \notin A_2$ and $c_2 \notin A_1$. Finally, the doctor is represented as an ambient d_B for some B with $a, b \in B$.

Initially, the ambient d_B is in the doctor's room dr_K , he then moves to the ward w_L and starts using services on the screen scr_{A_2} . The graphical representation of our setting is given in Figure 1. The ambients are represented by boxes, whereas dashed lines represent the communication channels. Next, we define our agents as follows:

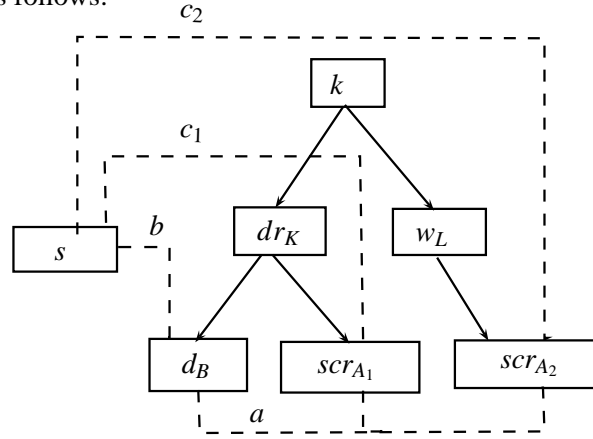


Figure 1: Intelligent Hospital setting

Agents *Server* and *S* are given below, where l is a finite sequence of values, v_1, v_2, \dots, v_k , for some k :

$$\text{Server}(v : l) \stackrel{\text{def}}{=} b(x). \text{if } (x = dr_K \text{ then } \bar{c}_1(v). \text{Server}(l) \\ \text{else if } x = w_L \text{ then } \bar{c}_2(v). \text{Server}(l) \text{ else } \text{Server}(v : l))$$

$$\text{Server}(\varepsilon) \stackrel{\text{def}}{=} 0 \quad S \stackrel{\text{def}}{=} s[\text{Server}(l)]$$

Agents Screen_m and Scr_m for $m \in \{1, 2\}$, are defined as follows:

$$\text{Screen}_m \stackrel{\text{def}}{=} c_m(x). \bar{a}(x). \text{Screen}_m \quad \text{Scr}_m \stackrel{\text{def}}{=} scr_{A_m}[\text{Screen}_m]$$

The agent Scr_{A_m} receives an input x from the server on c_m and outputs x on a . Since $a \in B$, the agent *Doc*, defined below, is able to view x via port a .

Finally, we define agents *Doctor* and *Doc* as follows:

$$\begin{aligned} \text{Doctor}(p,l) &\stackrel{\text{def}}{=} \bar{b}(p).a(x).\text{Doctor}(p,x:l) \\ &\quad + \text{out } p.\bar{b}(k).\text{in } r.\bar{b}(r).a(x).\text{Doctor}(r,l) \quad p,r \in \{dr_K, w_L\} \text{ and } r \neq p \\ \text{Doc} &\stackrel{\text{def}}{=} d_B[\text{Doctor}(dr_K, \varepsilon)] \end{aligned}$$

We use p to represent the initial location of *Doc*, here $p = dr_K$. When *Doc* leaves p by performing *out* p capability, his new location becomes k . He now may enter r by *in* r , and send his location to *Server*. In this particular situation, $r = w_L$ since $r \neq p$ and $p = dr_K$.

The Intelligent Hospital system is represented by the parallel composition of the server and the building, which contains doctor's room, ward, the doctor and two screens:

$$S \mid k[dr_K[\text{Doc} \mid \text{Scr}_1] \mid w_L[\text{Scr}_2]]$$

For simplicity we assume that the server S sends only a single piece of information, namely $l = v : \varepsilon$ for some v . Initially *Doc* is in dr_K and S wants to send the value v to *Doc* via either Scr_1 or Scr_2 . There are two possible sequences of execution of the Intelligent Hospital system. These sequences are:

$$(i) \quad \xrightarrow{\tau_{b(dr)}} \xrightarrow{\tau_{c_1(v)}} \xrightarrow{\tau_{a(v)}} \quad (ii) \quad \xrightarrow{\tau_{out}} \xrightarrow{\tau_{b(k)}} \xrightarrow{\tau_{in}} \xrightarrow{\tau_{b(w)}} \xrightarrow{\tau_{c_2(v)}} \xrightarrow{\tau_{a(v)}}$$

In the first sequence, *Doc* sends its location dr_K to S on port b , the server in response sends v to Scr_1 on port c_1 , and then *Doc* views v via port a . These interactions are indicated by appropriate labels that annotate the τ s of this sequence. In the second case, *Doc* leaves the dr_K and enters the ward by its *out* dr_K and *in* w_L capabilities. It sends its current location to S on port b after executing every move capability. The server in response sends v to the Scr_2 on port c_2 , and then the screen displays v to *Doc* on port a .

4 Adding Context-Awareness

In this section we extend the calculus even further by adding a context-awareness mechanism. In smart indoor settings, location is considered an important entity for providing communication among various portable and static structures. We consider location as one of the most typical forms of context, and propose a location-awareness feature, by introducing new constructs $ploc(x)$ and $sloc(x)$, that query an ambient's parent and sibling names respectively.

We add $ploc(x)$ and $sloc(x)$ to CMC_b , finally giving our full calculus CMC . The definition of μ in Table 2 is extended to include further $ploc(x)$ and $sloc(x)$. Also, the definition of λ in Table 2 is extended to include further auxiliary labels $plocI(z)$, $slocI(z)$ and $amb n_B$.

4.1 Reduction Semantics for CMC

The reduction semantics of CMC is given in terms of the structural congruence relation, \equiv , and the reduction relation, \rightarrow . The axioms for $ploc(x)$ and $sloc(x)$ are given in Table 6.

$$\begin{aligned} m_A[n_B[ploc(x).P \mid Q] \mid R] &\rightarrow m_A[n_B[P\{x \leftarrow m_A\} \mid Q] \mid R] && \text{(Red Ploc)} \\ m_A[P] \mid n_B[sloc(x).Q \mid S] &\rightarrow m_A[P] \mid n_B[Q\{x \leftarrow m_A\} \mid S] && \text{(Red Sloc)} \end{aligned}$$

Table 6: Reduction axioms for $ploc$ and $sloc$

Structural congruence, \equiv , for CMC processes is as in Section 2 where capabilities C include additionally $ploc(x)$ and $sloc(x)$. The reduction relation, \rightarrow , for CMC processes is as in Definition 3 except that it satisfies additionally the axioms in Table 6.

$$\begin{array}{c}
\text{(Act-Ploc)} \frac{}{ploc(x).P \xrightarrow{ploc(z)} P\{x \leftarrow z\}} \quad (z \text{ doesn't appear in } P) \\
\text{(Ploc1)} \frac{}{n_B[P] \xrightarrow{ploc1(z)} \langle n_B[P] \rangle 0} \quad \text{(\tau-Ploc)} \frac{P \xrightarrow{ploc1(z)} \nu \tilde{p}\langle P' \rangle Q \quad P' \xrightarrow{ploc(z)} P''}{m_A[P] \xrightarrow{\tau} (\nu \tilde{p})m_A[P''\{z \leftarrow m_A\} | Q]} (**).
\end{array}$$

Table 7: SOS rules for $ploc$. Condition (**) is as in Table 4

4.2 SOS Semantics for $ploc$ and $sloc$

The SOS rules for $ploc(x)$ and $sloc(x)$ in Tables 7, 8 and 5. As before, we use concretions in our rules. We illustrate reductions and transitions associated with the $ploc(x)$ capability by considering $m_A[n_B[ploc(x).P_1 | P_2] | Q]$, where n_B is a child of m_A and P_1, P_2 and Q are some processes. We assume for simplicity that P_2 and Q have no private names. The construct $ploc(x)$ enables n_B to find out the name of its parent (here m_A) and pass it to P via x . By the reduction rule Red Ploc we get $m_A[n_B[ploc(x).P_1 | P_2] | Q] \longrightarrow m_A[n_B[P_1\{x \leftarrow m_A\} | P_2] | Q]$, where $P_1\{x \leftarrow m_A\}$ denotes process P_1 with all occurrences of x replaced by m_A .

Now we show how to use τ -Ploc in Table 7. The τ -Ploc rule uses the notion of *lookahead* as, for example, in [24]. In order to derive a τ -transition of $m_A[P]$ we need to ensure that P contains an ambient enquiring parent's name. This is achieved by $P \xrightarrow{ploc1(z)} \nu \tilde{p}\langle P' \rangle Q$ where P' contains this ambient. The agent P' then perform $ploc(z)$ to substitute the parent's name: $P' \xrightarrow{ploc(z)} P''$. Hence P' is used both on the right-hand side and on the left-hand side of the premises in τ -Ploc, so τ -ploc has a lookahead.

Now, to derive the τ -transition of $m_A[n_B[ploc(x).P_1 | P_2] | Q]$ we must identify the ambient enquiring parent's name. To achieve this we introduce a new action $ploc1(z)$ and by Ploc1 we obtain $n_B[ploc(x).P_1 | P_2] \xrightarrow{ploc1(z)} \langle n_B[ploc(x).P_1 | P_2] \rangle 0$. By Par-Ploc1 we have

$$n_B[ploc(x).P_1 | P_2] | Q \xrightarrow{ploc1(z)} \langle n_B[ploc(x).P_1 | P_2] \rangle (0 | Q), \text{ where } z \notin fn(Q) \quad (\text{A})$$

Transition A matches the first premise of τ -Ploc for the agent $m_A[n_B[ploc(x).P_1 | P_2] | Q]$.

Now, $n_B[ploc(x).P_1 | P_2]$ must be able to perform the capability $ploc(x)$, thus giving the right-hand side premise of τ -Ploc:

$$n_B[ploc(x).P_1 | P_2] \xrightarrow{ploc(z)} n_B[P_1\{x \leftarrow z\} | P_2], \text{ where } z \notin fn(P_2) \quad (\text{B})$$

Since we have A and B, by τ -Ploc we obtain

$$\begin{aligned}
m_A[n_B[ploc(x).P_1 | P_2] | Q] &\xrightarrow{\tau} m_A[n_B[P_1\{x \leftarrow z\} | P_2]\{z \leftarrow m_A\} | 0 | Q] \\
&\equiv m_A[n_B[P_1\{x \leftarrow z\} | P_2]\{z \leftarrow m_A\} | Q]
\end{aligned}$$

Since z does not appear free in P_2 by rules for substitution, the transition is as required: $m_A[n_B[ploc(x).P_1 | P_2] | Q] \xrightarrow{\tau} m_A[n_B[P_1\{x \leftarrow m_A\} | P_2] | Q]$

We now consider the correspondence of the reduction semantics and the operational semantics for CMC. Let T''' be a sub-calculus of CMC that consists of all operators of CMC apart from the prefixing with actions (including τ) operators, the choice operator and the relabelling operator. We easily have the soundness part of this correspondence between the two semantics:

Theorem 3. $\forall P, P' \in T'''. P \rightarrow P' \implies \exists Q \in T'''. P \xrightarrow{\tau} Q \equiv P'$.

We conjecture that the completeness part of the correspondence between the operational semantics and reduction semantics is also valid. The proof relies on several auxiliary lemmas. For example, if $\lambda = ploc1(z)$ then the lemma for λ is:

$$\begin{array}{c}
\text{(Act-Sloc)} \frac{}{sloc(x).P \xrightarrow{sloc(z)} P\{x \leftarrow z\}} \quad (z \text{ doesn't appear in } P) \\
\text{(Sloc1)} \frac{}{m_A[P] \xrightarrow{sloc1(z)} \langle m_A[P] \rangle 0} \quad \text{(Sib-Amb)} \frac{}{n_B[P] \xrightarrow{amb\ n_B} P} \quad \text{(Par-Amb)} \frac{P \xrightarrow{amb\ n_B} P'}{P \mid Q \xrightarrow{amb\ n_B} P'} \\
\text{(\tau-Sloc)} \frac{P \xrightarrow{sloc1(z)} \nu \tilde{p} \langle P' \rangle P''' \quad P' \xrightarrow{sloc(z)} P'' \quad Q \xrightarrow{amb\ n_B} Q'}{P \mid Q \xrightarrow{\tau} \nu \tilde{p} (P'' \{z \leftarrow n_B\} \mid P''')} (*)
\end{array}$$

Table 8: SOS rules for *sloc*. Condition (*) is as in Table 4

Lemma 5. If $P \xrightarrow{ploc1(z)} (\nu \tilde{p}) \langle P' \rangle P'''$ and $P' \xrightarrow{ploc(z)} P''$, where variable z does not appear in P , then $P \equiv \nu \tilde{p} (n_B[ploc(x).P_1 \mid P_2] \mid P_3)$, $P' \equiv n_B[ploc(x).P_1 \mid P_2]$, $P''' \equiv P_3$ and $P'' \equiv n_B[P_1\{x \leftarrow z\} \mid P_2]$ for some P_1, P_2, P_3, n_B with $n_B \notin \tilde{p}, z \notin fn(P_2) \cup fn(P_3)$ and \tilde{p} a set of private ambient names in P .

5 Interactive Shopping Mall Case Study

This case study illustrates the usefulness of global communication, *in* n_B , *out* n_B , and *ploc*(x) features of CMC. The shopping mall consists of a number of retail outlets, clients and personal digital assistants (PDAs). To offer clients a high level of services, there is a server that delivers services to clients on requests via PDAs which are distributed inside the mall. The tree representation of the shopping mall setting is given in Figure 2, where the initial setting is given on the left-hand side and the final setting is on the right hand side. In this figure, the ambient *sm* is the shopping mall with two retail outlets *m*

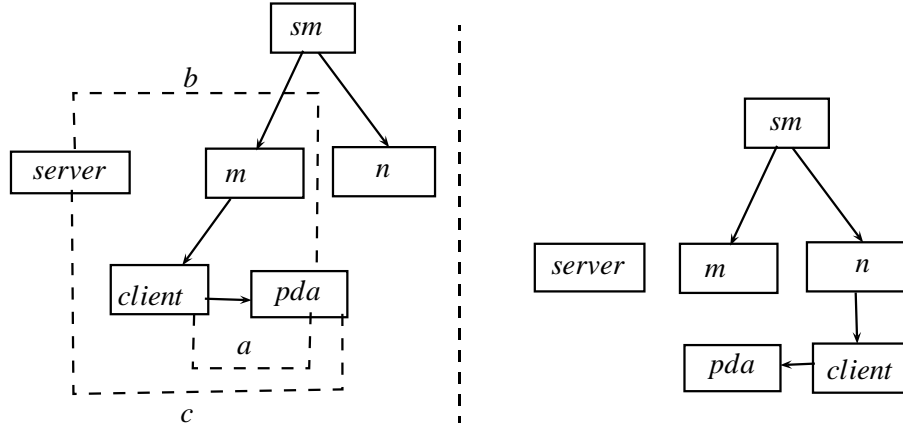


Figure 2: Interactive Shopping Mall settings

and n . For simplicity we have only one client and one PDA, represented by the ambients *client* and *pda* respectively, which are inside *m*.

Scenario: The client wishes to move from her current location m to a target location n inside the mall. She picks up a *pda* and sends the two locations to the *server* and requests for the path from m to n . The server generates this path as a sequence of capabilities and delivers it to the *client* via *pda*.

We define our setting as follows where C' , P' and S' are some processes:

$$\begin{aligned}
& \nu abc (sm[m[client[ploc(x).\bar{a}(x,n).a(u).u.C' \mid pda[a(y_1,y_2).\bar{b}(y_1,y_2).c(z).\bar{a}(z).P']]] \mid n[] \mid \\
& \quad server[b(x_1,x_2).\bar{c}(path(T,x_1,x_2)).S']]
\end{aligned}$$

Here, $path(T, x_1, x_2)$ is a function that calculates a path between the source node x_1 and the target node x_2 in a given tree T . The only possible execution sequence from this state is $\xrightarrow{\tau_{ploc}} \xrightarrow{\tau_a} \xrightarrow{\tau_b} \xrightarrow{\tau_c} \xrightarrow{\tau_a} S''$ for some S'' . In this sequence *client* acquires parent's name by $ploc(x)$ and sends her source and the target locations to *server* via a . The *server* in response calculates the $path(m, n)$ between the two locations and delivers it back to the *client*. In this particular case, the path calculated from m to n is *out m.in n*. Now the system has the form $S'' \equiv vabc(sm[m[client[out m.in n.C' | pda[P']]] | n[] | server[S'])$. After executing *out m.in n* the final state of the system becomes $vabc(sm[m[] | n[client[C' | pda[P']]] | server[S'])$, and is represented on the right hand side of Figure 2.

6 Conclusion and Related Work

We have proposed CMC for the modelling of mobility, communication and context-awareness in the setting of ubiquitous computing. The notion of ambients mobility has been modelled by the *in n_B* and *out n_B* capabilities [4]. A new form of global communication has been introduced in CMC which is similar to that in Milner's CCS. Ambient's name has been tagged with the set of ports which are functioning as a restriction on global communication, specified at the level of ambients. A labelled transition system semantics has been developed, where $P \xrightarrow{\tau} Q$ represents not only a binary communication of processes as in CCS but also the ambients' mobility steps by means of their *in n_B* and *out n_B* capabilities. This has been achieved by additional labels and specialised transitions from processes to the so-called outcomes which are either processes or concretions.

Recently, a number of variants of MA have been introduced. *Boxed Ambients (BA)* [2] inherits mobility primitives, namely the *in* and *out* capabilities from Mobile Ambients and introduce a direct communication method between parent and child. *Channel Ambient calculus (CA)* [18] is a variant of Boxed Ambients. In CA, channels are defined as a first class objects and the communication is either between parent and child or between siblings. To the best of our knowledge, the ambient calculi do not support a direct interaction of an agent with a subagent inside another agent. Communication can only happen between the two adjacent agents, namely communication between parent and child or between siblings. CMC has introduced a new form of global communication by defining ambients as $m_A[P]$, where m is the name of the ambient, A is the set of ports that m is allowed to communicate on, and P is an executing agent.

Poslad in [20] addressed a number of theoretical concepts in the context of ubiquitous computing. In ubiquitous computing setting computations could be mobile and context-aware as, for example, in [17, 21]. Satoh has researched spatial organisation of systems [22, 23] and concluded that technological advancements have enabled computing devices to become aware of their surroundings. Location-awareness has turned out to be useful in many applications, in particular, in determining position, navigation, tracking, and monitoring of ubiquitous computing devices. The notion of bigraph has been introduced by Milner in [16] with the idea of presenting two independent structures on the same set of nodes. A bigraph is a mathematical structure consisting of a place graph and a link graph with common nodes. Process calculi and behavioural equivalences have led to an approach in bigraph theory somewhat different from the well-known tradition of graph rewriting [15]. Leonhardt [9] classified location models into geometric and symbolic models. In geometric models locations are represented as coordinates systems, whereas symbolic location models use the notion of place and labelling the locations. We use the notion of place to model location, and represent the structure of our system by a hierarchical space tree. The nodes represent the places, objects or computing devices, whereas the edges represent the containment relations between objects. Each node or object is represented by named ambient, which may

contain nested ambients inside, as in [4].

A *Calculus of Context Aware Ambients (CCA)* [5] describes the context-awareness requirements of the mobile systems. It introduces the notion of context expression that constraints the capabilities. We also add a context-awareness mechanism to our calculus by introducing two capabilities to it. The new capability $ploc(x).P$ allows an ambient to acquire the name of its parent and pass it as x to P , whereas $sloc(x).P$ enquires the sibling's name of an ambient. Conversation Calculus [25, 3] is designed for expressing and analysing service based systems. It proposes a spatial communication topology where conversation contexts are used as message exchange patterns. The construct $here(x)$ that allows access to the conversation medium in Conversation Calculus is similar to the $ploc(x)$ and $sloc(x)$ capabilities of our calculus. These capabilities are not precisely used for only communication, whereas in Conversation Calculus conversation contexts are proposed as communication medium that controls information sharing among processes. Sessions [8] introduce a communication context among various partners to exchange messages based on previously agreed scheme, and sessions of specific patterns are introduced to express communication primitives. In CMC, we have modelled physical contexts and have intuitively used ambients to represent the structures. The systematic addition of context-awareness primitives smoothly increases the expressiveness power of the calculus.

In past few years, several operational semantics have been developed for MA and its variants as, for example, in [10, 12, 11]. The authors in [10] introduce a labelled transition system based operational semantics, and a labelled bisimulation equivalence which is proved to coincide with reduction barbed congruence. We also develop a labelled transition semantics and prove that the semantics coincides with the standard reduction semantics. Our labelled transition semantics is inspired by that in [10]. The main difference is that we do not use the co-capabilities, hence preserving the standard MA semantics. We have defined barbed bisimulation and congruence, and capability barbed bisimulation and congruence and have showed that the respective congruence relations of the two forms of barbs coincide. The notion of behavioural equivalence and the proof method for establishing the equivalence is inspired by that in [10]. The authors in [10], use co-actions and passwords that help them in proving their results, whereas the use of global communication in CMC is fundamental in proving the results. The expressiveness and usefulness of the calculus has been illustrated by presenting intelligent hospital and interactive shopping mall case studies, where the relevant constructs are used to model various features of the calculus.

References

- [1] Bergstra, J. A., and Klop, J. W. (1984): *Process Algebra for Synchronous Communication*. *Information and Control* 60(1-3), pp. 109–137, doi:10.1016/S0019-9958(84)80025-X. Available at [http://dx.doi.org/10.1016/S0019-9958\(84\)80025-X](http://dx.doi.org/10.1016/S0019-9958(84)80025-X).
- [2] Bugliesi, M., Castagna, G. and Crafa, S (2004): *Access control for mobile agents: The calculus of boxed ambients*. *ACM Transactions on Programming Languages and Systems* 26(1), pp. 57–124, doi:10.1145/963778.963781. Available at <http://doi.acm.org/10.1145/963778.963781>.
- [3] Caires, L. and Vieira, H. T. (2010): *Analysis of Service Oriented Software Systems with the Conversation Calculus*. LNCS, Springer, pp. 6–33, doi:10.1007/978-3-642-27269-1_2.
- [4] Cardelli, L, Gordon, Andrew D (2000): *Mobile Ambients*. *Theoretical Computer Science* 240(1), pp. 177–213, doi:10.1016/S0304-3975(99)00231-5.
- [5] François S., Cau, A. and Zedan, H. (2011): *The Calculus of Context-aware Ambients*. *Journal of Computer and System Science* 77(4), pp. 597–620, doi:10.1016/j.jcss.2010.02.003.
- [6] Gul, N. (2015): *A Calculus of Mobility and Communication for Ubiquitous Computing*. Ph.D. thesis, University of Leicester, England.

- [7] Hoare, C. A. R. (1985): *Communicating Sequential Processes*. Prentice-Hall.
- [8] Honda, K. and Vasconcelos, V. T. and Kubo, M. (1998): *Language Primitives and Type Discipline for Structured Communication-Based Programming*. In: *Programming Languages and Systems, LNCS 1381*, Springer, pp. 122–138, doi:10.1007/BFb0053567.
- [9] Leonhardt, U. (1998): *Supporting Location-Awareness in Open Distributed Systems*. Ph.D. thesis, Imperial College London.
- [10] Merro, M. and Hennessy, M. (2006): *A bisimulation-based semantic theory of Safe Ambients*. *ACM Transactions on Programming Languages and Systems* 28(2), pp. 290–330, doi:10.1145/1119479.1119482.
- [11] Merro, M. and Nardelli, F. Z. (2003): *Bisimulation Proof Methods for Mobile Ambients*. In: *ICALP, LNCS*, Springer, pp. 584–598, doi:10.1007/3-540-45061-0_47.
- [12] Merro, M. and Nardelli, F. Z. (2005): *Behavioral theory for mobile ambients*. *Journal of ACM* 52(6), pp. 961–1023, doi:10.1145/1101821.1101825. Available at <http://doi.acm.org/10.1145/1101821.1101825>.
- [13] Milner, R (1989): *Communication and Concurrency*. Prentice Hall Europe.
- [14] Milner, R (1999): *Communicating and Mobile Systems: The π -calculus*. Cambridge University Press.
- [15] Milner, R. (2008): *Bigraphs and Their Algebra*. *Electronic Notes Theoretical Computer Science* 209, pp. 5–19, doi:10.1016/j.entcs.2008.04.002. Available at <http://dx.doi.org/10.1016/j.entcs.2008.04.002>.
- [16] Milner, R (2009): *The Space and Motion of Communicating Agents*. Cambridge University Press.
- [17] Olaru, A. and Gratie, C. (2011): *Agent-Based, Context-Aware Information Sharing for Ambient Intelligence*. *International Journal on Artificial Intelligence Tools* 20(6), pp. 985–1000, doi:10.1142/S0218213011000498.
- [18] Phillips, A., Yoshida, N. and Eisenbach, S (2004): *A Distributed Abstract Machine for Boxed Ambient Calculi*. In: *Programming Languages and Systems, LNCS 2986*, Springer, pp. 155–170, doi:10.1007/978-3-540-24725-8_12. Available at http://dx.doi.org/10.1007/978-3-540-24725-8_12.
- [19] Plotkin, G. D (2004): *A structural approach to operational semantics*. *Journal of Logic and Algebraic Programming* 60-61, pp. 17–139.
- [20] Poslad, S (2009): *Ubiquitous Computing: Smart Devices, Environments and Interactions*, 1st edition. Wiley.
- [21] Ranganathan, A. and Campbell, A. H. (2003): *An infrastructure for context-awareness based on first order logic*. *Personal and Ubiquitous Computing* 7(6), pp. 353–364, doi:10.1007/s00779-003-0251-x.
- [22] Satoh, I (2005): *A location model for pervasive computing environments*. In: *Pervasive Computing and Communications, IEEE International Conference on*, IEEE Computer Society, pp. 215–224, doi:10.1109/PERCOM.2005.3.
- [23] Satoh, I (2005): *A Spatial Model for Ubiquitous Computing Services*. In: *IEICE Transactions on Communications*, E88-B, pp. 923–931, doi:10.1093/ietcom/e88-b.3.923. Available at <http://dx.doi.org/10.1093/ietcom/e88-b.3.923>.
- [24] Ulidowski, I. (1992): *Equivalences on Observable Processes*. In: *Proceedings of the Seventh Annual Symposium on Logic in Computer Science, Santa Cruz, California, USA, June 22-25, 1992*, IEEE Computer Society, pp. 148–159, doi:10.1109/LICS.1992.185529. Available at <http://dx.doi.org/10.1109/LICS.1992.185529>.
- [25] Vieira, H. T., Caires, L. and Seco, J. C. (2008): *The Conversation Calculus: A Model of Service-Oriented Computation*. *LNCS 4960*, Springer, pp. 269–283, doi:10.1007/978-3-540-78739-6_21. Available at http://dx.doi.org/10.1007/978-3-540-78739-6_21.
- [26] Weiser, M (1999): *The computer for the 21st century*. *Scientific American* 3, pp. 3–11, doi:10.1145/329124.329126. Available at <http://doi.acm.org/10.1145/329124.329126>.
- [27] Weiser, M (1999): *Some Computer Science Issues in Ubiquitous Computing*. In: *Communications of the ACM*, 3, pp. 75–84, doi:10.1145/329124.329127. Available at <http://doi.acm.org/10.1145/329124.329127>.

A SOS Rules for communication

$$\begin{array}{l}
\text{(Input)} \frac{}{a(z).P \xrightarrow{a(v)} P\{v/z\}} \quad (v \in V) \\
\text{(Res-Act)} \frac{P \xrightarrow{\alpha} P'}{(va)P \xrightarrow{\alpha} (va)P'} \quad (a \notin \text{fn}(\alpha)) \\
\text{(Par-Com)} \frac{P \xrightarrow{a(x)} P' \quad Q \xrightarrow{\bar{a}(x)} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'} \\
\text{(Rel)} \frac{P \xrightarrow{\alpha} P'}{P[f] \xrightarrow{f(\alpha)} P'[f]} \\
\text{(Struct)} \frac{P \equiv Q \quad Q \xrightarrow{l} Q' \quad Q' \equiv P'}{P \xrightarrow{l} P'} \\
\text{(Output)} \frac{}{\bar{a}(x).P \xrightarrow{\bar{a}(x)} P} \\
\text{(Sum)} \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'} \\
\text{(Par-Act)} \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q} \\
\text{(Const)} \frac{P \xrightarrow{\alpha} P'}{A \xrightarrow{\alpha} P'} \quad (A \stackrel{\text{def}}{=} P)
\end{array}$$