# A Calculus for Mobile Network Systems

Jinfeng Huang[1], Ad Verschueren[2], Henri Aalderink[3], and Johan Lukkien[4]

[1] Eindhoven University of Technology, EESI,
5600MB Eindhoven, the Netherlands
J.Huang@tue.nl
[2] Eindhoven University of Technology, Electrical Engineering Department,
5600MB Eindhoven, the Netherlands
[3] TNO Industrial Technology,
5600HE Eindhoven, the Netherlands
[4] Eindhoven University of Technology, Mathematics and Computer Science Department,
5600MB Eindhoven, the Netherlands

**Abstract.** The rapidly increasing demand for ubiquitous communication has led to the widespread use of wireless networks. These systems consist of a group of independently executing components which may migrate through some space during the course of the computation, and the connectivity between the components changes with their migration [13][14]. This new communication paradigm, the distribution-dependent communication model, cannot be directly supported by the communication model of the $\pi$-calculus [1][5][6][16]. In this paper, we first analyze the communication features of mobile network systems. Sequentially, we propose a two-layer connection model for the communication between components in mobile network systems. Finally, an extension of the $\pi$-calculus , entity calculus, is introduced. The entity calculus can also act as a starting point for a designer-friendly approach for the modelling of mobile network systems by incorporating it in tooling support.

## 1 Introduction

Formal methods refer to the use of mathematic frameworks to model, calculate, analyze and predict target systems. In principle, formal methods bring a set of eagerly wanted merits: unambiguity, clarity, conciseness and consistency which are helpful for verifying correctness of system properties and finding design faults at the earliest possible design stages [15]. Process algebras, such as CCS [9] and its successor the $\pi$-calculus [10], have been demonstrated very efficient for verifying interactive systems such as wired communication systems and mobile agent systems at the system level, and the formal languages based on these algebras have gained great success in modelling such systems.

However, the new communication paradigm of mobile network systems, the distribution-dependent communication model, cannot be directly supported by the communication model of the $\pi$-calculus or most other process calculi. This argument comes from the following simple example:

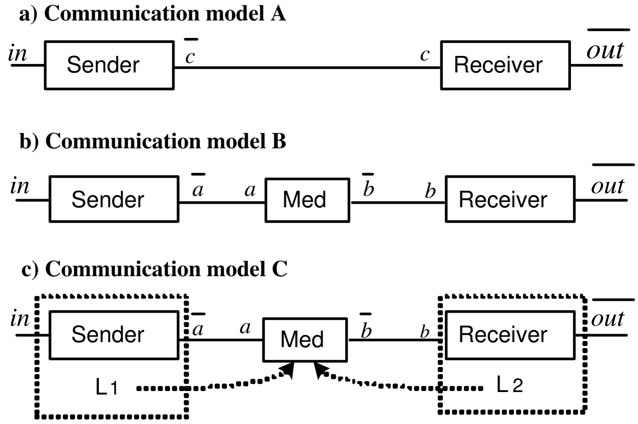


**Fig. 1.** Communication Models

Fig.1a shows the synchronous communication model of the $\pi$-calculus using a simple ideal communication system. It is described in the $\pi$-calculus as follows:

$System \stackrel{def}{=} (Sender|Receiver)$

$Sender \stackrel{def}{=} in.\overline{c}.Sender$

$Receiver \stackrel{def}{=} c.\overline{out}.Receiver$

In Fig. 1b, we insert a $Med$ between the $Sender$ and the $Receiver$ to represent the non-ideal communication channel where messages are transmitted. We assume that the behavior of the $Med$ is independent from the $Sender$ and the $Receiver$ . The system is given as follows:

$System \stackrel{def}{=} (Sender\{a/c\}|Med|Receiver\{b/c\})$

$Med \stackrel{def}{=} a.(\overline{b}.Med + Med)$

The $Med$ receives a message from the $Sender$ and may or may not forward it to the $Receiver$. This communication model is general enough to represent all kinds of wired communication paradigms. However, in mobile network systems, the media between the $Sender$ and the $Receiver$ is not independent from them any more. The location (or other physical states) of processes may affect the interaction results between them (see Fig 1c). In this paper, we denote these physical states of processes as the external state which indicates the physical distribution of processes.

It is technically difficult to model the dynamic physical distribution of processes in $\pi$-calculus and most other process calculi. Furthermore, the verification of these models based on these calculi often neglects the physical distribution effect, which is an important feature of mobile network systems. Most of the properties of mobile network systems, such as the presence of deadlock states, are relevant to the resource allocation strategy of systems, and the dynamic physical distribution is one of the most important factors that lead to resource reallocation. Section 5 of this paper gives an example to show that a deadlock-free model may present deadlock states when extended with a dynamic physical distribution.

Since the physical distribution of systems imposes a considerable effect on the interaction between the processes, the main motivation of the entity calculus is to integrate the dynamic physical distribution into the calculus. This is also quite useful for "executable models" of mobile network systems where the designer can check the properties of a system based on different dynamic physical distributions.

The content of this paper is organized as follows. In section 2, we present related work in the context of process calculi. In section 3, we analyze the mobility issue in mobile network systems; the entity calculus is introduced in section 4. In section 5, a simple case study is presented to demonstrate the effect of the dynamic distribution. Finally, we summarize the work and give a direction for future research.

## 2 Related Work

Mobility of concurrent distributed systems has received much attention in the last ten years, and several calculi have been proposed to model mobile systems. According to different aims, these calculi can be categorized into two important subgroups: one is aimed at examining the causality which states the action sequence of concurrent processes; the other is focused on investigating the locality which addresses the distribution of systems [16]. Both approaches investigate different aspects of concurrent systems.

The $\pi$-calculus is a typical causality calculus whose dynamic reconfiguration is expressed by passing the name of a channel between processes through communication. This is a flexible and moderately simple way to express dynamic connections [10][17]. However, the $\pi$-calculus can not directly support the modelling of the dynamic physical distribution of systems.

Aceto [1] and Boudol et al [5] first try to combine processes with a location attribute using different ways and extend CCS with different location bisimulations. Later, Sangiorgi [16] makes the same extension to the $\pi$-calculus based on similar notations. These "locality calculi" focus on the location-sensitive equivalence and treat the locality and the causality as two orthogonal concepts. However, This assumption fails to hold in mobile network systems because the interaction between components of mobile network systems are influenced by locations (or other physical states) of processes(see Fig. 1c).

Fournet's distributed join-calculus [6][7] attempts to provide language support for asynchronous, distributed, and mobile programming. A simple failure model is provided to detect the failure of communications due to locations of processes. The $\pi$F calculus [3][4] derived from the $\pi$-calculus is more closely related to our work. The authors, Ando et al, define a field concept which is a set of locality constraints on communications; they also model the environment independently.

The motivation of our work is to present a generalized calculus for mobile network systems based on the former work. This calculus incorporates the dynamic distribution of systems, which allows the designer to have the capability of verifying system characteristics such as absence of deadlock, in the scope of different dynamic distributions.

## 3 Mobility in Mobile Network Systems

Mobile network systems are composed of a group of independently executing components which may migrate through some space during the course of computation. The connectivity between the components may change with their migration [13][14]. Every component in mobile network systems may exhibit two behaviors: one is migration in some space which is denoted as the physical behavior; the other is independent computation and synchronous communication which is denoted as the logical behavior. The physical behavior and the computation behavior of components may cause the physical mobility and the logical mobility respectively.

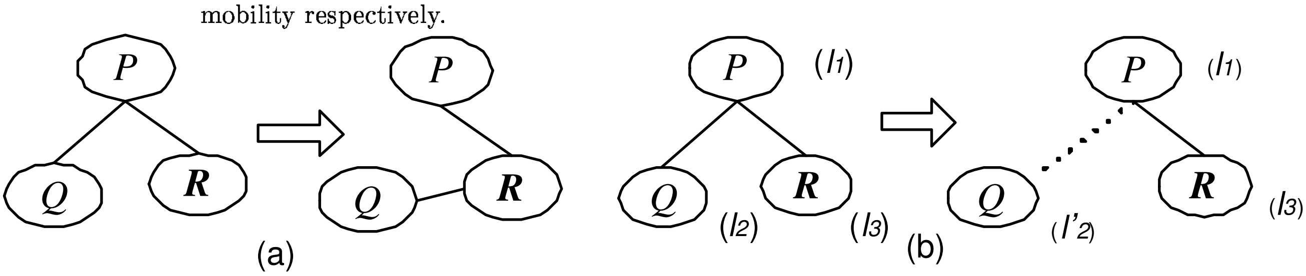


**Fig. 2.** Two kinds of Mobility
($a$) Logical Mobility: $R$ changes its interaction relation with $P$ and $Q$. The topology of the processes is changed. ($b$) Physical Mobility: $l_1$, $l_2$, $l_3$ and $l'_2$ are physical locations of the processes; $Q$ changes its physical location from $l_2$ to $l'_2$. The topology of the processes may not be changed.

The mobility concept has been widely used, but the meaning of mobility is a bit vague. The $\pi$-calculus adopts "links movement in the virtual space of linked processes" as its mobility concept [10]. This captures logical mobility: the dynamic process topology (see Fig. 2a). We add to this the concept of physical mobility (see Fig. 2b): "physical devices movement in the physical space of physically linked devices". The link concepts in these two kinds of mobility are not the same; the former represents the interaction relationship between computation processes and is called logical communication channel; the latter addresses physical constraints on the communication between components. It is an enabler for the former. In the following parts of this section, we investigate the roles of different factors in mobile networks systems, and examine how both kinds of mobility act on each other.

1. *The External State:* Abstraction of physical attributes, such as physical location, antenna height, power level, sensitivity, etc.
2. *The Global Environment:* A function determining whether a physical channel can be formed by two given external states.
3. *The Logical Channel:* A channel through which two processes can interact with each other.
4. *The Physical Channel:* A path through which physical signals can be transmitted from one terminal to another. It puts the constrains on the logical channel.

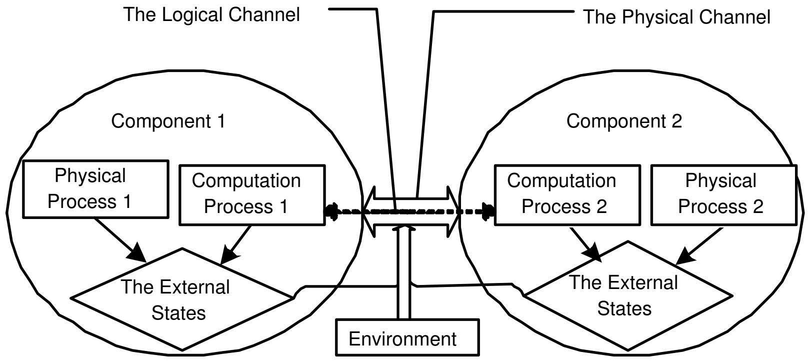


**Fig. 3.** The Basic Elements of Mobile Network Systems

In $\pi$ -calculus, a logical channel is represented by a pair of names. It is not easy to use the same way to represent the physical channel. In our calculus the physical channel will be determined by the external states of both components and the environment. Changes of the physical connections of components can be regarded as changes of the external states of components.

A typical component in mobile network systems may consist of three parts: one or more physical processes, one or more computation processes and the external state. Both kinds of processes can change the external state. A computation process residing in a component can interact with other computation processes through a logical channel. If two computation processes reside in different components, the interaction between them will be constrained by the physical channel between the two components. In this paper, we denote this interaction as external-state-constrained synchronized communication. If there is a physical channel between two components, the communication between the two computation processes is synchronized in which the receiver gets messages from the sender; otherwise, it is still a synchronous communication, but the receiver does not get any message from the sender and still keeps its state.

The physical mobility is triggered by changes of the external state. The logical mobility is always invoked by interactions between the computation processes. The physical mobility may cause some expired logical channels go into effect and at the same time may lead to the expiration of some available logical channels. Note that this change of expiration state of the logical channel itself is not logical mobility. In mobile systems, the expiration state of the logical channel will affect the interaction result between the computation processes.

## 4 The Entity Calculus

In the context of process calculi, locality and causality are treated as somehow orthogonal concepts in most literature [1][5][16]. However, in mobile network systems, the distributions of systems are influenced by a great variety of physical states of mobile terminals such as power level, antenna height and location. These physical features of mobile terminals are not orthogonal to the action sequences of processes anymore. For example, in GSM network systems, a mobile terminal can adjust its power level according to its interaction with others, and in turn, its power level influence the interaction with others terminals. In our calculus, we generalize the locality of processes to the external state of processes. The external state puts constraints on the communication between processes.

### 4.1 Basic Concepts

Syntax of the entity calculus is based on that of the standard $\pi$-calculus, but the external state for processes is extended and used for describing the dynamic physical distribution of systems. Several basic concepts are introduced first.

**Basic Entity** A basic entity consists of two components: a group of processes and an external state.

For example, $\{P, q\}$ is a simplest entity representation where $P$ is a process, $q$ is its external state and they are bounded by a pair of braces.

Two parallelism granularities, process-layer granularity and entity-layer granularity, can be checked in the entity calculus. In two entities $\{P|_p Q, p\}$ and $\{P, p\}|_e\{Q, q\}$, $|_p$ and $|_e$ represent the process parallelism and the entity parallelism respectively. Processes $P$ and $Q$ in the former expression run concurrently and synchronize in the same way as in the $\pi$-calculus; whereas in the latter one, the synchronization is affected by the physical channel between them.

**Constraint set $Cset$ and the Physical Channel** In the entity calculus, we model the global environment as a global constraint set $Cset$, which determines the existence of the physical channel between two entities having specific external states.

For example, if there are only three external states $p$, $q$, $r$ in the specification and $Cset$ is given as the following:

$Cset = \{(p,q),(q,p),(p,p),(q,q)\}$

$\{\alpha.P,p\}|_e\{\overline{\alpha}.Q,q\}$ and $\{\alpha.P,q\}|_e\{\overline{\alpha}.R,r\}$ are two possible entity combinations based on the above external states. The synchronization result of the former combination is $\{P,p\}|_e\{Q,q\}$ because $(p,q)$ belongs to $Cset$, whereas the synchronization result of the latter combination is $\{\alpha.P,q\}|_e\{R,r\}$, because there is no physical channel between two parallel entities ($(q,r) \notin Cset$).

### 4.2 Syntax of the Entity Calculus

The entity calculus supports process-layer and entity-layer parallelism. Syntax of the calculus is given in the following.

**Process-layer Syntax:**

1. State action set: $Sact \stackrel{def}{=} \{\widehat{\alpha},\widehat{\beta},\widehat{\gamma}...\}$.
2. Send action set: $Csact \stackrel{def}{=} \{\overline{\alpha},\overline{\beta}...\}$.
3. Receive action set: $Cract \stackrel{def}{=} \{\alpha,\beta,...\}$.
4. Process action: $\eta ::= \alpha\,(\widetilde{x}) \,|\, \overline{\alpha}\,\langle\widetilde{x}\rangle \,|\, \tau \,|\, \widehat{\alpha}$. The parallel communication actions at the process layer are represented by $\alpha\,(\widetilde{x})$ and $\overline{\alpha}\,\langle\widetilde{x}\rangle$, where $\overline{\alpha}\,\langle\widetilde{x}\rangle$ sends a message with a value list $\widetilde{x}$ and $\alpha(\widetilde{x})$ receives a message with a parameter list $\widetilde{x}$. $\tau$ is an unobservable action between processes. $\widehat{\alpha}$ is a state action which changes the shared external state of processes to $\widehat{\alpha}$.
5. Process: $P ::= NIL \,|\, \eta.P |\, Q|_pR \,|\, P{+}R \,|\, (x)Q \,|\, Q\,\langle\widetilde{x}\rangle \,|\, [x=y]Q$. Action prefix $\eta$ has one more choice than that of the standard $\pi$-calculus. $|_p$ is a process parallel operator. $Q|_pR$ is a process in which $P$ and $Q$ run concurrently and synchronize with each other by a pair of send and receive actions in the same entity.

**Entity-layer Syntax:**

1. Entity action: $\mu ::= \alpha\,(\widetilde{x})\,@\widehat{\beta} \,|\, \overline{\alpha}\,\langle\widetilde{x}\rangle\,@\widehat{\beta} \,|\, \tau@\widehat{\beta} \,|\, \widehat{\alpha}@\widehat{\beta} \,|\, \varepsilon$. The parallel actions at the entity layer are represented by $\alpha\,(\widetilde{x})\,@\widehat{\beta}$, $\overline{\alpha}\,\langle\widetilde{x}\rangle\,@\widehat{\beta}$, $\tau@\widehat{\beta}$, $\widehat{\alpha}@\widehat{\beta}$ and $\varepsilon$, where $\widehat{\beta}$ represents the external state of a basic entity. $\varepsilon$ is an internal action between entities. $\tau@\widehat{\beta}$ and $\varepsilon$ are two different unobservable actions of entity. The former represents the internal action in one basic entity and cannot be observed at the process layer. The latter represents the internal action between two different basic entities.
2. Entity: $A ::= \{P,p\} \,|\, A|_eB \,|\, (x)B \,|\, B\,\langle x\rangle$. $|_e$ is an entity parallel operator. $A|_eB$ is an entity in which $A$ and $B$ run concurrently and synchronize with each other by the external-state-constrained communication. The synchronization of processes in different entities are constrained by the physical channel between entities.

3. Constraint set $Cset$: $Cset \subseteq Sact \times Sact$. $Cset$ puts constraints on the interaction of processes belonging to different basic entities. For example: A and B are basic entities having external states $\widehat{\alpha}$, $\widehat{\beta}$ respectively. If B can receive messages from A, $(\widehat{\alpha}, \widehat{\beta})$ belongs to $Cset$.

**Conventions:**

1. $\widehat{\alpha}, \widehat{\beta}, \widehat{\gamma}$...range over state action set $Sact$.
2. $\overline{\alpha}, \overline{\beta}$... range over communication action set $Csact$.
3. $\alpha, \beta$... range over communication action set $Cract$.
4. Process action set $Pact ::= Csact \cup Cract \cup Sact \cup \{\tau\}$ .
5. $\eta$ represents the process action.
6. Entity action set $Eact ::= \{\eta @ \widehat{\alpha} | \eta \in Pact \; and \; \widehat{\alpha} \in Sact\} \cup \{\varepsilon\}$.
7. $\mu$ represents the entity action.
8. $P, Q, R$ range over process set $Proc$.
9. $A, B, C$ range over entity set $Entity$.

### 4.3 Structure Congruence

Two parallel elements, process and entity, are defined in syntax of the entity calculus. Accordingly, process-layer structure congruence and entity-layer structure congruence are given as follows.

**Process-layer Congruence:**[5]

(1) $P + Q \equiv_p Q + P$, $(P + Q) + R \equiv_p P + (Q + R)$;
(2) $P|_p Q \equiv_p Q|_p P$, $(P|_p Q)|_p R \equiv_p P|_p (Q|_p R)$, $P|_p NIL \equiv_p P$;
(3) $(x)NIL \equiv_p NIL$, $(x)(y)P \equiv_p (y)(x)P$,
$(x)(P|_p Q) \equiv_p P|_p (x)Q \; if \; x \notin fn(P)$[6];
(4) $Q \langle \widetilde{x} \rangle \equiv_p \{\widetilde{x}/\widetilde{y}\} P_Q \; where \; Q \langle \widetilde{y} \rangle \stackrel{def}{=} P_Q$.

**Entity-layer Congruence:**

(1) $A|_e B \equiv_e B|_e A$, $(A|_e B)|_e C \equiv_e A|_e (B|_e C)$, $A|_e \{NIL, p\} \equiv_e A$;
(2) $(x)(y)A \equiv_e (y)(x)A$, $(x)(A|_e B) \equiv_e A|_e (x)B \; if x \notin fn(A)$,
$(x)\{P, \widehat{\beta}\} \equiv_e \{(x)P, \widehat{\beta}\} \; where \; \widehat{\beta} \in Sact$;
(3) $\{P, \widehat{\beta}\} \equiv_e \{Q, \widehat{\beta}\} \; where \{P, \widehat{\beta}\} \equiv_e \{Q, \widehat{\beta}\} \; and \; \widehat{\beta} \in Sact$.

The last structure congruence shows the relation between process-layer congruence and entity-layer congruence.

---

[5] In the entity calculus, the restrictions only take effects on send names and receive names.

[6] $fn(P)$ represents all the free send and receive names of process $P$;

### 4.4 Reduction Rules

In the entity calculus, parallelism of two granularity levels is treated. Entities concurrently run and are synchronized through external-state-constrained communication. Inside the entity, processes are carried out in parallel and synchronized through communication channels. The processes bound in an entity share the same external state. Unlike other process calculi, we treat state actions as observable actions. The state actions change their shared external state, and the physical channels between entities are changed according to the constraint set.

Similar to the syntax of the entity calculus, the reduction rules of the entity calculus are also classified into two layers. The process-layer reduction rules are basically same as those of the standard $\pi$-calculus, while three additional reduction rules are introduced to express the relation of processes and their entities.

**Process-layer Reduction Rules:**

$$pPREFIX: \ \eta.P + M \xrightarrow{\eta} P$$

$$pLPAR: \ \frac{P \xrightarrow{\eta} P'}{P|_pQ \xrightarrow{\eta} P'|_pQ} \qquad pRPAR: \ \frac{Q \xrightarrow{\eta} Q'}{P|_pQ \xrightarrow{\eta} P|_pQ'}$$

$$pRES: \ \frac{P \xrightarrow{\eta} P'}{(x)P \xrightarrow{\eta} (x)P'} \ if \ \eta \neq x \qquad pCOMM: \ \frac{P \xrightarrow{\overline{\alpha}} P'; Q \xrightarrow{\alpha} Q'}{P|_pQ \xrightarrow{\tau} P'|_pQ'}$$

$$pe1: \ \frac{P \xrightarrow{\eta} P'}{\{P, \widehat{\beta}\} \xrightarrow{\eta@\widehat{\beta}} \{P', \widehat{\beta}\}} \ if \ \eta \in Csact \cup \tau$$

$$pe2: \ \frac{P \xrightarrow{\eta} P'}{\{P, \widehat{\beta}\} \xrightarrow{\eta@\widehat{\beta}} \{P + P', \widehat{\beta}\}} \ if \ \eta \in Cract$$

$$pe3: \ \frac{P \xrightarrow{\eta} P'}{\{P, \widehat{\beta}\} \xrightarrow{\eta@\widehat{\beta}} \{P', \eta\}} \ if \ \eta \in Sact$$

Rules $pe1$, $pe2$, and $pe3$ describe the relation between the process action and the entity action. $pe2$ denotes that there is a non-determination inside an entity when it issues a receive action.

**Entity-layer Reduction Rules:**[7]

$$eLPAR: \frac{A \xrightarrow{\mu} A'}{A|_e B \xrightarrow{\mu} A'|_e B} \qquad ePAR: \frac{A|_e B \xrightarrow{\mu} C}{B|_e A \xrightarrow{\mu} C}$$

$$eRES: \frac{A \xrightarrow{\eta @ \widehat{\beta}} A'}{(x)A \xrightarrow{\eta @ \widehat{\beta}} (x)A'} \; if \; x \neq \eta$$

$$eCOMM1: \frac{A \xrightarrow{\overline{\alpha} @ \widehat{\beta}} A'; \{P, \widehat{\gamma}\} \xrightarrow{\alpha @ \widehat{\gamma}} \{P' + P, \widehat{\gamma}\}}{A|_e\{P, \widehat{\gamma}\} \xrightarrow{\varepsilon} A'|_e\{P', \widehat{\gamma}\}} \; if \; (\widehat{\beta}, \widehat{\gamma}) \in Cset$$

$$eCOMM2: \frac{A \xrightarrow{\overline{\alpha} @ \widehat{\beta}} A'; \{P, \widehat{\gamma}\} \xrightarrow{\alpha @ \widehat{\gamma}} \{P' + P, \widehat{\gamma}\}}{A|_e\{P, \widehat{\gamma}\} \xrightarrow{\varepsilon} A'|_e\{P, \widehat{\gamma}\}} \; if \; (\widehat{\beta}, \widehat{\gamma}) \notin Cset$$

Rules $eCOMM1$, $eCOMM2$ together with $ePAR$ describe the external-state-constrained communication. If there exists a physical channel between two entities, the process in the receive entity goes into the next state. Otherwise, the process remains in its original state. These rules explain how the external state of a entity influence its interaction with other entities.

### 4.5 Bisimulation:

To discuss the equivalence of two granularity parallelism, we introduce two bisimulation relations. Process-layer bisimulation is used to check whether two processes are identical in the context of the entity calculus, and entity-layer bisimulation is used to distinguish two entities.

**Process Bisimulation:** A binary relation $S_p \subseteq Proc \times Proc$ over processes is a process bisimulation (or a $\approx_p$ bisimulation), if $(P, Q) \in S_p$ implies,
(a) Whenever $P \xrightarrow{\eta} P'$, then for some $Q'$ , $Q \xrightarrow{\eta} Q'$ and $(P', Q') \in S_p$ ;
(b) Whenever $Q \xrightarrow{\eta} Q'$, then for some $P'$ , $P \xrightarrow{\eta} P'$ and $(P', Q') \in S_p$ ;
where $\eta \in Pact$.

**Entity bisimulation:** A binary relation $S_e \subseteq Entity \times Entity$ over entities is a entity bisimulation (or a $\approx_e$ bisimulation), if $(A, B) \in S_e$ implies,
(a) Whenever $A \xrightarrow{\mu} A'$, then for some $B'$ , $B \xrightarrow{\mu} B'$ and $(A', B') \in S_e$ ;
(b) Whenever $B \xrightarrow{\mu} B'$, then for some $A'$ , $A \xrightarrow{\mu} A'$ and $(A', B') \in S_e$ ;
where $\mu \in Eact$.

The relation between process bisimulation and entity bisimulation is shown in proposition 1.

[7] We can use rules $\boldsymbol{eLPAR}$, $\boldsymbol{eCOMM1}$, $\boldsymbol{eCOMM2}$ together with $\boldsymbol{ePAR}$ to deduce their symmetrical rules. For example: $\boldsymbol{eLPAR}$ together with $\boldsymbol{ePAR}$ deduces to rule $\boldsymbol{eRPAR}$: $\frac{A \xrightarrow{\mu} A'}{B|_e A \xrightarrow{\mu} A'|_e B}$.

**Lemma 1.** $P_1 \approx_p Q_1$ *and* $P_2 \approx_p Q_2$ *imply* $P_1 + P_2 \approx_p Q_1 + Q_2$.

*Proof.* Whenever $P_1 + P_2 \xrightarrow{\eta} P'$ , $\eta$ action may be issued by $P_1$ or $P_2$.

Suppose $\eta$ action is issued by $P_1$. According to the definition of $\approx_p$, there exists some $Q'$, $Q_1 \xrightarrow{\eta} Q'$ and $P' \approx_p Q'$. We use $pPREFIX$ to deduce $Q_1 + Q_2 \xrightarrow{\eta} Q'$.

Similarly, if $\eta$ action is issued by $P_2$, there exists some $Q'$, $Q_1 + Q_2 \xrightarrow{\eta} Q'$ and $P' \approx_p Q'$.

Therefore, whenever $P_1 + P_2 \xrightarrow{\eta} P'$, there exists some $Q'$, $Q_1 + Q_2 \xrightarrow{\eta} Q'$ and $P' \approx_p Q'$.

Similarly we can prove whenever $Q_1 + Q_2 \xrightarrow{\eta} Q'$, there exists some $P'$, $P_1 + P_2 \xrightarrow{\eta} P'$ and $P' \approx_p Q'$.

According to the definition of $\approx_p$, $P_1 + P_2 \approx_p Q_1 + Q_2$.

**Proposition 1.** $P \approx_p Q$ *implies for all* $\widehat{\beta} \in Sact$, $\{P, \widehat{\beta}\} \approx_e \{Q, \widehat{\beta}\}$.

*Proof.* Construct a binary relation $S \stackrel{def}{=} \{(\{P, \widehat{\beta}\}, \{Q, \widehat{\beta}\}) \mid P \approx_p Q;\ P, Q \in Proc;\ \forall \widehat{\beta} \in Sact\}$.

Now, we prove relation $S$ is an entity bisimulation.

It can be easily seen that $\varepsilon$ can not be performed by a basic entity. Entity action $\mu$ for basic entities can be expressed as $\eta @ \widehat{\beta}$. According to the different action sets which $\eta$ belongs to, there are three different cases.

(1) Suppose $\eta \in Csact \cup \{\tau\}$. In this case, $P \xrightarrow{\eta} P'$. According to $pe1$, $\{P, \widehat{\beta}\} \xrightarrow{\eta @ \widehat{\beta}} \{P', \widehat{\beta}\}$. For $P \approx_p Q$, there exists some $Q'$, $P \approx_p Q$ and $Q \xrightarrow{\eta} Q'$. We use $pe1$ to deduce $\{Q, \widehat{\beta}\} \xrightarrow{\eta @ \widehat{\beta}} \{Q', \widehat{\beta}\}$. According to the definition of $S$, $(\{P', \widehat{\beta}\}, \{Q', \widehat{\beta}\}) \in S$.

(2) Suppose $\eta \in Cract$. In this case, $P \xrightarrow{\eta} P'$. According to $pe2$, $\{P, \widehat{\beta}\} \xrightarrow{\eta @ \widehat{\beta}} \{P' + P, \widehat{\beta}\}$. For $P \approx_p Q$, there exists some $Q'$, $P \approx_p Q$ and $Q \xrightarrow{\eta} Q'$. We use $pe2$ to deduce $\{Q, \widehat{\beta}\} \xrightarrow{\eta @ \widehat{\beta}} \{Q' + Q, \widehat{\beta}\}$. According to the definition of $S$ and Lemma 1, $(\{P' + P, \widehat{\beta}\}, \{Q' + Q, \widehat{\beta}\}) \in S$.

(3) Suppose $\eta \in Sact$. In this case, $P \xrightarrow{\eta} P'$. According to $pe3$, $\{P, \widehat{\beta}\} \xrightarrow{\eta @ \widehat{\beta}} \{P', \eta\}$. For $P \approx_p Q$, there exists some $Q'$, $P \approx_p Q$ and $Q \xrightarrow{\eta} Q'$. We use $pe3$ to deduce, $\{Q, \widehat{\beta}\} \xrightarrow{\eta @ \widehat{\beta}} \{Q', \eta\}$. According to the definition of $S$, $(\{P', \eta\}, \{Q', \eta\}) \in S$.

We use $A$ represents $\{P, \widehat{\beta}\}$ and $B$ represents $\{Q, \widehat{\beta}\}$. $A \xrightarrow{\mu} A'$, there exists some $B'$, $B \xrightarrow{\mu} B'$ and $\{A', B'\} \in S$

Similarly, we can prove whenever $B \xrightarrow{\mu} B'$, there exists some $A'$, $A \xrightarrow{\mu} A'$ and $\{A', B'\} \in S$.

Therefore, $S$ is an entity bisimulation, and the rest is straight forward.

### 4.6 Comparison of the entity calculus with the $\pi$F calculus

The $\pi$F calculus is proposed by Ando et al in [3][4]. Unlike other locality process calculi, it models the environment as an independent part, which simplifies the expression of mobile network systems in calculi. In the entity calculus, we adopt the same idea to model the environment independently. However, there are still several notable differences between the entity calculus and the $\pi$F calculus.

1. We generalize the locality concept to the external state to adapt to mobile network systems.
2. In most locality calculi, the locality of processes are treated orthogonal to their action sequences. In the $\pi$F calculus, the movement of processes is regarded as unobservable actions. We extend the locality of processes to the external state of processes, which is not independent from their action sequences. For example, mobile processes can change their power level state according to their interaction with other processes.
3. Congruence $\{P|_p Q, \widehat{\beta}\} \equiv_e \{P, \widehat{\beta}\}|_e\{Q, \widehat{\beta}\}$ is not satisfied any more in the entity calculus, because the external state is shared by all the processes in the entity and may be changed by any one of them.
4. The external-state-constrained communication specifies the communication between two entities when there is no physical channel between them (see rule $eCOMM2$). The $\pi$F calculus and other extensions of the $\pi$-calculus do not give an explicit rule to specify this kind of communication.

## 5 A Case Study

In this section, we give an example to show how a deadlock-free system may reach a deadlock state when we consider the dynamic distribution effect.

The whole system is composed of three mobile components, $C_1$, $C_2$ and $C_3$. $C_1$ can move in some physical space defined in $Sact$ & $Cset$. At the same time, $C_1$, $C_2$ and $C_3$ independently execute and synchronize with each other. Physical behavior and computation behavior exist in the system.

First, let us check the correctness of the system without considering the dynamic physical distribution. Let $P$, $Q$ and $R$ represent the computation behaviors of three mobile components respectively using the syntax of the $\pi$-calculus.

$P \overset{def}{=} \overline{\alpha}\langle\beta\rangle.r(x).P'$

$Q \overset{def}{=} \alpha(y).\overline{y}\langle b\rangle.Q'$

$R \overset{def}{=} \beta(t).\overline{\gamma}\langle c\rangle.R'$

$System \overset{def}{=} P|Q|R$

The deduction of the system is given as follows:

$P|Q|R \overset{def}{=} \overline{\alpha}\langle\beta\rangle.r(x).P'|\alpha(y).\overline{y}\langle b\rangle.Q'|\beta(t).\overline{\gamma}\langle c\rangle.R'$

$\xrightarrow{\tau} r(x).P'|\overline{\beta}\langle b\rangle.Q'|\beta(t).\overline{\gamma}\langle c\rangle.R'$

$\xrightarrow{\tau} r(x).P'|Q'|\overline{\gamma}\langle c\rangle.R'$

$\xrightarrow{\tau} P'|Q'|R'$

At the beginning, $P$ and $Q$ synchronize at action $\alpha$. $P$ sends a new channel name to $Q$, and $Q$ receives new channel name $\beta$ and dynamically builds a new channel with $R$. Then, $Q$ and $R$ synchronize with each other at action $\beta$, and $\overline{\gamma}\langle c\rangle.R'$ and $r(x).P'$ at action $\gamma$. After three synchronizations, $P$, $Q$, $R$ go into the states $P'$, $Q'$, $R'$ respectively.

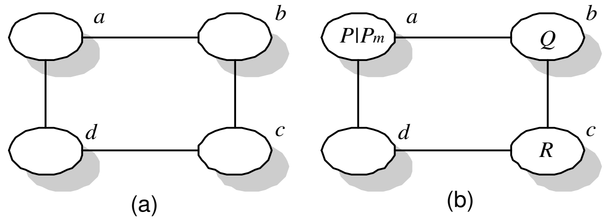


**Fig. 4.** The Physical Space and Entities in the System

Now, let us check the correctness of the system considering its dynamic physical distribution. First we define the physical space as shown in Fig.4a

$Sact \stackrel{def}{=} \{a, b, c, d\}$,

$Cset \stackrel{def}{=} \{(a, b), (b, a), (b, c), (c, b), (c, d), (d, c), (a, d), (d, a), (a, a), (b, b), (c, c), (d, d)\}$.

The physical movement of $C_1$ is given as the following:

$P_m \stackrel{def}{=} d, a.P_m$.

The system (Fig. 4b) is represented as the following:

$C_1 \stackrel{def}{=} \{P|P_m, a\}$,

$C_2 \stackrel{def}{=} \{Q, b\}$,

$C_3 \stackrel{def}{=} \{R, c\}$,

$System \stackrel{def}{=} C_1|C_2|C_3$.

The definitions of $P$, $Q$, $R$ are the same as in the previous example.

The deduction of the system is given as follows.

$C_1|C_2|C_3$

$\stackrel{def}{=} \{\overline{\alpha}\langle\beta\rangle.r(x).P'|_p d.a.P_m, a\}|_e\{\alpha(y).\overline{y}\langle b\rangle.Q', b\}|_e\{\beta(t).\overline{\gamma}\langle c\rangle.R', c\}$

$\stackrel{d@a}{\rightarrow} \{\overline{\alpha}\langle\beta\rangle.r(x).P'|_p a.P_m, d\}|_e\{\alpha(y).\overline{y}\langle b\rangle.Q', b\}|_e\{\beta(t).\overline{\gamma}\langle c\rangle.R', c\}$

$\stackrel{\varepsilon}{\rightarrow} \{r(x).P'|_p a.P_m, d\}|_e\{\alpha(y).\overline{y}\langle b\rangle.Q', b\}|_e\{\beta(t).\overline{\gamma}\langle c\rangle.R', c\}$

At first, $C_1$ moves from state $a$ to $d$; then $P$ tries to synchronize with $Q$ at action $\alpha$. But there are is no physical channel between $C_1$ and $C_2$ according to $Cset$, using rule $eCOMM2$, $C_1$ goes into $\{r(x).P'|_p a.P_m, d\}$, and $C_2$ goes into $\{\alpha(y).\overline{y}\langle b\rangle.Q', b\}$. At this moment, the system goes into deadlock state in which all of the three computation processes are waiting for receiving messages.

In this example, we demonstrate that a mobile network system may exhibit different behaviors according to its physical distribution. The main motivation of the entity calculus is trying to reason about this situation, and provide a mathematic framework for mobile network system modelling and verification.

## 6 Conclusion

In this paper, we proposed a concurrent calculus, entity calculus, for formalizing mobile network systems. In the entity calculus, an entity represents a group of processes that share the same external state. In this way, the external-state dependent communication paradigm can be more easily modelled in the entity calculus than in the $\pi$- calculus.

The future work may involve the incorporation of the entity calculus into the work of the POOSL group at Eindhoven University. The group has developed a formal language POOSL [12]that is based on the CCS framework and extended this framework with real time and probability features [8][18][19]. The effect of dynamic physical distribution on mobile network specifications will be analyzed based on a formal model of the GSM hand-over procedure.

## 7 Acknowledgments

Here I would like to express my gratitude to Jeroen Voeten and Ana Sokolova for their helpful comments on an earlier draft of this paper.

This research has been sponsored through the PROGRESS organization by STW, Philips and TNO.

## References

1. Acteo,L.: A static view of localities. Formal Aspects of Computing, 6:201–222, (1994)
2. Amadio, R.M..: An synchronious model of locality, failure, and process mobility. In Proceedings of COORDINATION'97, LNCS 1282, 374–391. Springer-Verlag, (1997)
3. Ando, T.; Takahashi, K.; Kato, Y.; Shiratori, N.: A Concurrent Calculus with Geographical Constraints. IEICE TRANS.FUNDAMENTALS, VOL.E81-A, NO.4, (1998).
4. Ando, T.; Takahashi, K.; Kato, Y.; Shiratori, N.: Maintenance of mobile system ambients using a process calculus. Computer Networks 32(2): 229-256 (2000)
5. Boudol, G.; Castellani, M.; Hennessy, M.; Kiehn, A.: A theory of processes with localities. Formal Aspects of Computing, 6:165–200, (1994)
6. Fournet, C.: The join calculus: a language for distributed moible programming. European Research Seminar on Advances in Distributed Systems (ERSADS'01), Italy (2001)
7. Fournet, C.; Gonthier, G.; Lévy, J-J. Maranget, L.; Rémy, D.: A calculus of mobile agents. Lecture Notes in Computer Science, Volume 1119, Springer Verlag (1996)
8. Geilen, M.C.W.; Voeten, J.P.M.; van der Putten, P.H.A.; van Bokhoven, L.J.; Stevens, M.P.J.: Object-Oriented Modeling and Specification using SHE. Journal of computer languages, special issue for VFM'99, Vol.27, Issue 1-3 (2001)

9. Milner, R.: Communication and Concurrency. International Series in Computer Science, Prentice Hall (1989)
10. Milner, R.: Communicating and mobile systems: the $\pi$-calculus. Cambridge University Press (1999)
11. Orava, F.; Parrow, J.: An algebraic verification of a mobile network. Formal Aspects of Computing 497–543 (1992)
12. Putten, P.H.A.; Voeten, J.P.M.: Specification of Reactive Hardware/Software Systems. PhD. thesis, Department of Electrical Engineering, Eindhoven University of technology, Eindhoven, The Netherlands (1997)
13. Roman, G.-C.; McCann, P.J.: A Notation and Logic for Mobile Computing. Formal Methods in System Design, 1-22. Kluwer Academic Publishers (1999)
14. Roman, G.-C.; Picco, G.; Murphy, A.: Software Engineering for Mobility: A Roadmap. Future of Software Engineering, 241-258. In Finkelstein, A, Editor. ACM Press (2000)
15. Rushby, J.: Formal Methods and their Role in the certication of Critical Systems. Technical Report CSL-95-1. Computer Science Laboratory, SRI International (1995)
16. Sangiorgi, D.: Locality and Non-interleaving Sematics in claculi for mobile processes. International Conference on Theoretical Aspects of Computer Software (TACS'94), Japan (1994)
17. Sangiorgi, D.; Walker, D.: The Pi-calculus: A Theory of Mobile Processes. Cambridge University Press (2001), ISBN 0521781779
18. Theelen, B.D.; Voeten, J.P.M.; van Bokhoven, L.J.; van der Putten, P.H.A.; de Jong, G.G.; Niemegeers, A.M.M.: Performance Modeling in the Large: A Case Study. In: Proceedings of the European Simulation Symposium (ESS) 174–181 (2001), ISBN 9077039023
19. Voeten, J.P.M.; van der Putten, P.H.A.; Geilen, M.C.W.; Stevens, M.P.J.: Towards System Level Performance Modeling. In: Proceedings of the 9th annual workshop on Circuits, Systems and Signal Processing: 593–507, STW/IEEE (1998)