A Calculus for Mobile Network Systems

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

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

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 [5][6]. The support for this physical-distribution-constrained communication paradigm cannot be explicitly seen in the communication model of the π-calculus [7]. 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 π-calculus, entity calculus is introduced.

1 Introduction

Process algebras, such as CCS [3] and its successor the π-calculus [4], have shown to be very effective 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.

In mobile network systems (later referred as MNS), the physical location (or other physical states such as power strength) of processes may affect the interaction results between processes. It is technically difficult to model the dynamic physical distribution of processes in the π-calculus. Furthermore, the verification of the π-calculus based models often neglects the physical distribution effect, which is an important feature of MNS. Most of the properties of MNS, 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.

Mobility of concurrent distributed systems has received much attention in the last ten years, and several calculi based on the π-calculus have been proposed to model the physical distribution properties. The πF calculus [1][2] derived from the π-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 incorporate the dynamic physical distribution into the calculus, which allows the designer to have the capability of verifying system characteristics such as the absence of a deadlock in the scope of different dynamic distributions.

The content of this paper is organized as follows. In section 2, we analyze the mobility issue in MNS; the entity calculus is introduced in section 3. Finally, we summarize the work.

2 Mobility in Mobile Network Systems

MNS 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 [5][6]. Every component in MNS may exhibit two behaviors: one is migration in some space, the other is independent computation and synchronous communication.

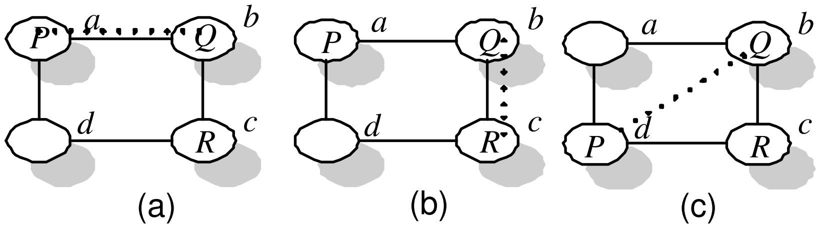

Fig. 1. Two kinds of Mobility
$P \overset{def}{=} \overline{\alpha}\langle\beta\rangle.0;\ Q \overset{def}{=} \alpha(y).\overline{y}\langle t\rangle.0;\ R \overset{def}{=} \beta(x).0$

In Fig 1, there are four physical locations: a, b, c, d. The solid lines represent physical signal reachability between physical locations and the dotted lines stand for the logical synchronous relation between processes. In this case, process P sends a new port name β to process Q, then Q uses the new port to send a name t to process R. After two synchronization, all of processes successfully terminate and reach the state 0.

Logical mobility: After P sends β to Q, the logical topology of the processes is changed from Fig 1(a) to Fig 1(b), and their physical distribution remains unchanged.

Physical mobility: After process Q moves from a to b, the physical distribution of processes is changed from Fig 1(a) to Fig 1(c). However, their logic topology may not be changed.

3 The Entity Calculus

In this section, we give an informal explanation of our formalization and then follows its syntax and its operational semantics. Our formalization is based on the standard π-calculus, while the physical distribution is added as an extension.

A basic entity consists of two components: a group of processes and an external state. For example, $\{P, q\}$ is a basic entity representation where P is a process or a group of processes, and q is its physical state. They are bounded by a pair of braces.

In entity calculus, the physical space is defined as $Sact$, which includes all possible physical states. $Cset$ is used to express the relation between physical states. For example, in Fig 1, $Cset$ is specified as follows:

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

In the calculus, parallelism of two granularity levels is treated. Entities concurrently run and synchronize through physical-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 physical state. Unlike other process calculi, we treat physical state actions as observable actions. The physical state actions change their physical states, which put constraints on their synchronization . These constraints are mainly caused by the fact that physical communication signals, being the carrier of logical messages, can not be exchanged between all of the physical states in the physical space.

Syntax of the Entity Calculus: The process-layer syntax is the same as that of the standard π-calculus except that the action set $Pact$ is extended with physical state actions. In entity calculus, we partition the action set into three independent action sets: physical state action set $Sact$, send action set $Csact$ and receive action set $Cract$.

The entity-layer syntax is given as follows:

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 ε, 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 $\widetilde{x}$. $\widehat{\beta}$ represents the external state of a basic entity. ε is an internal action between entities. $\tau@\widehat{\beta}$ and ε 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: $A ::= \{P,p\} \,|\, A|_e B \,|\, (x)B$. $|_e$ is an entity parallel operator, and $|_p$ is a process parallel operator. $(x)B$ makes action name x local to entity **B**.
3. Constraint set $Cset$: $Cset \subseteq Sact \times Sact$.

Structure Congruence: Two parallel elements, process and entity, are defined in the syntax of the entity calculus. Accordingly, the process-layer structure congruence ($\equiv_p$) and the entity-layer structure congruence ($\equiv_e$) are defined. Again, the process-layer structure congruence is the same as that of the standard π-calculus. The smallest entity-layer structure congruence is given as follows: [5]

(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) $(x)(y)A \equiv_e (y)(x)A$, $(x)(A|_e B) \equiv_e A|_e(x)B\ if x \notin fn(A)$[6],
$(x)\{P,\widehat{\beta}\} \equiv_e \{(x)P,\widehat{\beta}\}\ where\ \widehat{\beta} \in Sact$;

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

[6] $fn(A)$ represents all the free send and receive names of entity A.

(3) $\{P, \widehat{\beta}\} \equiv_e \{Q, \widehat{\beta}\}$ *where* $P \equiv_p Q$ *and* $\widehat{\beta} \in Sact$.

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

Operational semantics: Similar to the syntax of the entity calculus, the operational rules of the entity calculus are also classified into two layers. The process-layer operational rules are basically the same as those of the standard π-calculus, while three additional rules, $r1$, $r2$, and $r3$, are introduced to express the relation of process transitions and entity transitions.

$$r1: \frac{P \xrightarrow{\eta} P'; \eta \in Csact \cup \tau}{\{P, \widehat{\beta}\} \xrightarrow{\eta @ \widehat{\beta}} \{P', \widehat{\beta}\}};\; r2: \frac{P \xrightarrow{\eta} P'; \eta \in Cract}{\{P, \widehat{\beta}\} \xrightarrow{\eta @ \widehat{\beta}} \{P + P', \widehat{\beta}\}};\; r3: \frac{P \xrightarrow{\eta} P'; \eta \in Sact}{\{P, \widehat{\beta}\} \xrightarrow{\eta @ \widehat{\beta}} \{P', \eta\}}$$

$r2$ denotes that there is a non-determination inside an entity when it issues a receive action. The selected branch is determined by its physical state and its peer's physical state, not by itself alone (see rules $eCOMM1$ and $eCOMM2$).

The entity-layer operational rules are given as follows:[7]

$$eLPAR: \frac{A \xrightarrow{\mu} A'}{A|_e B \xrightarrow{\mu} A'|_e B};\; ePAR: \frac{A|_e B \xrightarrow{\mu} C}{B|_e A \xrightarrow{\mu} C};\; eRES: \frac{A \xrightarrow{\eta @ \widehat{\beta}} A'; x \neq \eta}{(x)A \xrightarrow{\eta @ \widehat{\beta}} (x)A'}$$

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

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

Rules $eCOMM1$, $eCOMM2$ together with $ePAR$ describe the physical-state-constrained communication. The result of synchronization depends on their physical states. If the physical signal can be transferred between two physical states, the process in the receive entity goes into the next state. Otherwise, the process remains in its original state.

Now, let us check the interaction between processes considering their dynamic physical distribution in Fig 1. The physical movement between location d and a is specified by process P_m ($P_m \stackrel{def}{=} d.a.P_m$).

The system is represented as the following:

$System \stackrel{def}{=} \{P|_p P_m, a\}|_e\{Q, b\}|_e\{R, c\}$.

The deduction of the system is given as the following.

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

$System \overset{def}{=} \{\overline{\alpha}\langle\beta\rangle.0|_p d.a.P_m, a\}|_e\{\alpha(y).\overline{y}\langle t\rangle.0, b\}|_e\{\beta(x).0, c\}$
$\overset{d@a}{\rightarrow} \{\overline{\alpha}\langle\beta\rangle.0|_p a.P_m, d\}|_e\{\alpha(y).\overline{y}\langle t\rangle.0, b\}|_e\{\beta(x).0, c\}$
$\overset{\varepsilon}{\rightarrow} \{0|_p a.P_m, d\}|_e\{\alpha(y).\overline{y}\langle t\rangle.0, b\}|_e\{\beta(x).0, c\}$

At first, P moves from location a to d; then P tries to synchronize with Q at action α. Using rule $eCOMM2$, P issues a send action and goes into terminate state, but Q does not receive the new port name from P. At present, both Q and R are waiting for receiving a message, thus the system goes into a deadlock state.

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.

4 Conclusion

In this paper, we proposed a concurrent calculus, entity calculus, for formalizing MNS. In the entity calculus, an entity represents a group of processes that share the same external state. In this way, the physical-distribution-constrained communication paradigm can be more easily modelled in the entity calculus than in the π- calculus.

5 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. Ando, T.; Takahashi, K.; Kato, Y.; Shiratori, N.: A Concurrent Calculus with Geographical Constraints. IEICE TRANS.FUNDAMENTALS, VOL.E81-A, NO.4, (1998).
2. Ando, T.; Takahashi, K.; Kato, Y.; Shiratori, N.: Maintenance of mobile system ambients using a process calculus. Computer Networks 32(2): 229-256 (2000)
3. Milner, R.: Communication and Concurrency. International Series in Computer Science, Prentice Hall (1989)
4. Milner, R.: Communicating and mobile systems: the π-calculus. Cambridge University Press (1999)
5. Roman, G.-C.; McCann, P.J.: A Notation and Logic for Mobile Computing. Formal Methods in System Design, 1-22. Kluwer Academic Publishers (1999)
6. 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)
7. Sangiorgi, D.: Locality and Non-interleaving Sematics in claculi for mobile processes. International Conference on Theoretical Aspects of Computer Software (TACS'94), Japan (1994)