

並行計算 πF を用いたモバイルシステム管理

安藤敏彦, 高橋薫, 加藤靖

仙台電波高専

〒 989-3124 仙台市青葉区上愛子字北原 1

Tel:022-391-5606, Fax.:022-391-6144

{tando,kato}@info.sendai-ct.ac.jp, kaoru@cc.sendai-ct.ac.jp

概要

本研究では、並行計算 πF を用いて、モバイルシステムの安定な運用を行うための方法を考察する。モバイルシステムの特徴は、通信主体の移動、および動的リンク切替である。 πF は、そのような移動性に加え、置かれている環境によるシステム挙動への制約を考慮した π 計算の拡張版である。本研究では、事故などの環境の変化から生じるシステムのトラブルを環境レベルで補償するために、通信主体間の接続性に注目する。この接続可能性に基づき、 πF でモデル化した環境を修復する指針を示す。

キーワード

モバイルシステム, πF , 地理的制約, 環境, 接続性

Mobile Systems Maintenance Using Concurrent Calculus πF

Toshihiko ANDO, Kaoru TAKAHASHI and Yasushi KATO

Sendai National College of Technology

1, Kitahara, Kami-Ayashi, Aoba-ku, Sendai

ABSTRACT

We consider a maintenance method for mobile systems using a concurrent calculus πF in this paper. πF is an extension of π -calculus, which is a calculus proposed to express effects of an environment into the behavior of a mobile system. The goal of this research is the repair of mobile systems at the environment level in case that an accident breaks down that environment. We show a guideline for repairing the damaged environment based on connectivity among communication entities.

keywords

mobile systems, πF , geographical constraints, ambient, connectivity

1 はじめに

近年、携帯電話や PHS などの移動電話、携帯端末の普及は著しい。これによって、「いつでも、どこでも」ネットワーク環境を利用することが可能になりつつある。また、移動可能なエージェント同士がネゴシエーションして、情報探索、商取引を行うシステムも実現しつつある。その他、これらの技術の発展とともに、高度道路交通システム (Intelligent Transport System, ITS) の研究も進んでいる。このような構成要素が物理的に移動し、かつ通信接続も動的に行われるようなシステム、いわゆるモバイルシステムは、有線ネットワークとはまた異なった特性を有している。モバイルシステムの最も大きな特徴は、(1) 通信主体が物理的に移動すること、(2) 移動に伴ってリンクの切り替えが動的になされることである。それらの特徴を的確に仕様で反映させ、それをシステム的设计や管理に利用できることが望まれている。

これらの移動性、動的リンクを数学的手法で厳密に記述するために、CHAM (Chemical Abstract machine) [1] など多くの形式記述技法が考えられている。その中でも特に、Milner の π 計算 [2] が有力である。プロセス代数ではアクション系列でプロセスの動作を記述するが、 π 計算はアクション名を値として他のプロセスに渡すことを許すので、動的なリンクを陽に記述することができる。ただし、次のような問題がある。 π 計算ではどんなプロセスの間でもコミュニケーションができると仮定している。しかし、実際の移動電話などでは端末の送信電力を押さえるために、セルシステムを採用しているなど、プロセス間のコミュニケーションが局所的であることが多い。そのため、 π 計算にそのような局所性を導入するよう様々な拡張が行われている。局所性を表現するためにプロセスの位置を明記した方法 [3]、補助的関数を使って接続の失敗を検出する方法 [4] が提案されている。また、 π 計算以外にもプロセス間の置かれている位置の間に距離を導入し、プロセス間のコミュニケーションに優先度を与えている研究も行われている [5]。

一般に、移動通信や ITS では建物の影響で通信が影響を受けるなど、環境による地理的制約がシステムの動作に大きく影響を及ぼしている。ところが、上述の方法ではそれらを考慮していない。そこで著者らはモバイルシステムに影響を与える環境を表現するため、場の概念を導入することで、 π 計算の拡張を行い、並行計算 πF を提案した [6]。これによって、環境の影響を表現できるとともに、プロセス記述が簡潔になった。環境の変化に対するモバイルシステムの適正な運用を行うために πF が利用できると思われる。ITS の通信プロトコルも πF で

記述できる [7]。

本研究では、環境が事故などによって小規模に変化した場合、それによってシステムの受ける障害を、環境レベルで補修する指針を示す。ここではプロセス自体が正常にもかかわらず環境の不具合で異常が生じる場合、環境を整備することでシステムを正常に保たせることを目的としている。環境は上述した場で表現されるが、このモデル上でプロセス間の「接続性」を保つために場をどのように修正すればよいかを示す。

2 並行計算 πF の概要

本節では、著者らが提案しているプロセス計算の一つである πF を説明する。プロセス計算では、プロセスと呼ばれる動作の単位がゲート (チャンネル) を介してコミュニケーションを行うと考えている。 π 計算では、この時にチャンネル名をデータとして送信することを許し、新しいチャンネルを発生させる。

$$\begin{aligned} \overline{call}[no].0 \mid call(x).\bar{x}.0 \mid no.0 &\rightarrow 0 \mid \bar{no}.0 \mid no.0 \\ &\rightarrow 0 \mid 0 \mid 0 \end{aligned}$$

この式は、チャンネル $call$ を介して、チャンネル名 no を渡し、それによって新しいチャンネル no が生成されることを示している。 π 計算では、この名前渡し (name-passing) によって、動的な接続を表現することができる。

ただし、一つ問題がある。 π 計算のコミュニケーションはグローバルに行われ、プロセスがどんな位置にいても考慮されない。しかし、移動通信では建築物の影響で通話が妨げられることが多い。そのため、移動通信システムなどのモバイルシステムではその管理において、システムの置かれている環境を把握していることが重要である。そこで、本研究では、このような環境によるコミュニケーションへの制約を「場 (field)」として表現し、それを π プロセスのパラメータとして付随させる。場は有向グラフで表現され、データの送られる方向を定めている。

定義 1 環境 (ambient)

環境はコミュニケーションに制約を与える場 F と、プロセスの移動 (movement) MV の組

$$A = \langle F, MV \rangle$$

からなる。場は次のように定義される。

$$F = \langle Loc, RL \rangle$$

ここで、 Loc は位置 (location) の集合、 $RL \subseteq Loc \times Loc$ はデータの送信方向を示す道 (road) の集合である。また、 MV は移動 (P, l, A) を元として持つ集合として与え

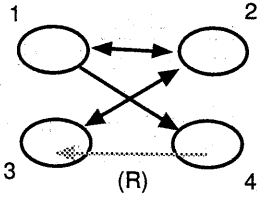


図 1: 環境の例: A_1

られる。ここで、 $P, l, A (\neq l)$ はそれぞれプロセス、位置、位置の集合であり、 (P, l, A) は P が l にいるとき A に含まれる位置のいずれかに移動できることを示している。

図 1 に、環境の例 A_1 を示す。この例では、場 \mathcal{F}_1, MV_1 は次のように与えられる。

$$\begin{aligned} \mathcal{F}_1 &= \langle Loc_1, RL_1 \rangle \\ Loc_1 &= \{1, 2, 3, 4\} \\ RL_1 &= \{(1, 4), (1, 2), (2, 1), (2, 3), (3, 2)\}. \\ MV_1 &= \{(R, 4, \{3\})\}. \end{aligned}$$

次に、 πF の構文と導出規則を示す。

定義 2 πF の構文

$$\begin{aligned} \text{名前関数} &: \underline{func} &&: Ch^n \rightarrow Ch^m; \underline{func}(\vec{b}) = \vec{c} \\ \text{名前} &: a &&:: a \mid \underline{func}(\vec{b}) \\ \text{アクション} &: \alpha &&:: a(\vec{b}) \mid \bar{a}[\vec{b}] \\ \text{プロセス} &: P &&:: 0 \mid \alpha.P \mid P+Q \mid P|Q \\ & && \mid \nu \vec{c}P \mid D(\vec{b}) \\ \text{レイバー} &: L &&:: \{P\}l \mid L|M \mid \nu \vec{c}L \end{aligned}$$

ここで、 Ch はチャンネル名全体の集合である。 $\bar{a}[\vec{b}]$ は、チャンネル a を通して、値 (名前) の列 \vec{b} を出力するアクションを示し、 $a(\vec{c})$ は、チャンネル a から値 (名前) を入力し、 \vec{c} はその値 (名前) に置き換えられることを示す。プロセスの構文は、左から停止、アクション前置、選択、並列接続、名前の隠蔽、値の代入を意味する。 πF 式 (レイバー (labor)) は、プロセス記述に位置を付加したものである。これによって、そのプロセスがどの位置にいるかを示す。例えば、 $\{P\}l$ はプロセス P が位置 l にいることを示す。

πF プロセスの構文は、マッチング ($[a = b]P$) が無いことや名前関数の存在を除けば、標準的な π 計算とほぼ同様である。合同関係も同様である。また、名前関数を用いると記述の抽象度を上げることができるので、記述が簡潔になる。

定義 3 導出規則

$$\begin{aligned} \text{COMM:} & \frac{\# \vec{b} = \# \vec{c}, (l, l_\lambda) \in RL \text{ for } \lambda \in \Lambda, \Delta (\neq \emptyset) \subseteq \Lambda}{\{\dots + \bar{a}[\vec{b}].P\}l \mid \prod_{\lambda \in \Lambda} \{\dots + a(\vec{c}).Q_\lambda(\vec{u})\}l_\lambda} \\ & \frac{\{P\}l \mid \prod_{\lambda \in \Lambda} \{Q_\lambda(\vec{b})\}l_\lambda \mid \prod_{\lambda' \in \Lambda - \Delta} \{\dots + a(\vec{c}).Q_{\lambda'}(\vec{u})\}l_{\lambda'}}{\{P\}l} \\ \text{MOVE:} & \frac{(P, l, A) \in MV, m \in A}{\{P\}l \xrightarrow{\tau} \{P\}m} \\ \text{PAR:} & \frac{L \xrightarrow{\zeta} L'}{L|M \xrightarrow{\zeta} L'|M} \\ \text{STRUCT:} & \frac{M \equiv L, L \xrightarrow{\zeta} L', L' \equiv M'}{M \xrightarrow{\zeta} M'} \\ \text{RES1:} & \frac{L \xrightarrow{a \circ Pls} L', x \neq a}{\nu x L \xrightarrow{a \circ Pls} \nu x L'} \\ \text{RES2:} & \frac{L \xrightarrow{a \circ Pls} L', x = a}{\nu x L \xrightarrow{\tau} \nu x L'} \end{aligned}$$

$L \xrightarrow{a \circ Pls} L'$ は、 L がチャンネル a を介して行われるアクションによって L' に遷移することを表す。 Pls はそのアクションが発生する位置の集合である。 $COMM$ において、 $\prod_{\lambda \in \Lambda} L_\lambda$ は $L_{\lambda_1} | L_{\lambda_2} | \dots$ ($\lambda_1, \lambda_2, \dots \in \Lambda$) である。 Λ はこの時点でコミュニケーション可能なプロセスのインデックス集合である。

プロセスは、それに付随する位置によってコミュニケーションが制限される ($COMM$)。 πF では、この規則によって、コミュニケーションの地理的制約を表現している。通常のプロセス計算では、peer-to-peer 通信に限定しているが、上述の $COMM$ では、ブロードキャストやマルチキャストも認めている。また、 $MOVE$ はプロセスの移動を表現している。

例 1 πF 式の導出例を示す。 no という番号を持つ携帯電話に電話の呼び出しをするプロセスを示す。プロセス P, Q, R を次のように与える。

$$\begin{aligned} P &\stackrel{\text{def}}{=} \overline{\text{call}}[no]. 0 + \overline{\text{break}}. 0, \\ Q &\stackrel{\text{def}}{=} \text{call}(x). \bar{x}. 0 + \overline{\text{break}}. 0, \\ R &\stackrel{\text{def}}{=} no. 0 + \overline{\text{break}}. 0. \end{aligned}$$

P は呼び出し側、 Q は基地局、 R は携帯電話を表す。このとき、次の πF 式 (レイバー) $L_1 = \{P\}1 \mid \{Q\}2 \mid \{R\}4$ は環境 A_1 の上で次のように導出される。

$$\begin{aligned} &\{P\}1 \mid \{Q\}2 \mid \{R\}4 \\ \xrightarrow{\text{call}@ \{1,2\}} &\{0\}1 \mid \{\bar{no}. 0\}2 \mid \{R\}4 \quad (COMM) \\ \xrightarrow{\tau} &\{0\}1 \mid \{\bar{no}. 0\}2 \mid \{R\}3 \quad (MOVE) \\ \xrightarrow{\text{no}@ \{2,3\}} &\{0\}1 \mid \{0\}2 \mid \{0\}3 \quad (COMM) \end{aligned}$$

この場合、最初 R は Q とは直接コミュニケーションできないが、 P がチャンネル a を通し R が利用できるチャネ

ル名 b を Q に渡し、かつ R が位置 '4' から '3' へ移動することで、 Q と R のコミュニケーションが可能になる。これで、 P から R への呼び出しが完了する。 □

例 2 上の P, Q, R を別の環境 $\mathcal{A}_2 = \langle \mathcal{F}_2, MV_2 \rangle$ に置いている (これを L_2 とする)。これは、 \mathcal{A}_1 からいくつかの道が除かれたものである。

$$\begin{aligned} \mathcal{F}_2 &= \langle Loc_2, RL_2 \rangle \\ Loc_2 &= \{1, 2, 3, 4\} = Loc_1 \\ RL_2 &= \{(1, 2), (1, 4)\}. \\ MV_2 &= \{(R, 4, \{3\})\} = MV_1. \end{aligned}$$

同じ初期位置から始めてみると、 P から Q へはコミュニケーションできるが、それ以降 Q から R へのコミュニケーションは行なえない。つまり、 P から R への呼び出しは中断される。

$\{0\}1 \mid \{Q\}2 \mid \{R\}4 \xrightarrow{call@1,2} \{0\}1 \mid \{\bar{no}.0\}2 \mid \{R\}4$
ここで用いた \mathcal{F}_2 は、何らかの原因で基地局からの電波が遮られていることに対応しており、それが携帯電話の呼び出しを妨げている。 □

3 接続性

前節の 2 つの例はプロセス自身は正常にもかかわらず、それが置かれる環境によってはシステムとして正常に働く場合と働かない場合があることを示している。これはプロセスの「接続性(connectivity)」の観点から整理することができる。

接続性を議論するために、「マーキング(marking)」操作を導入する。コミュニケーション毎にマーキングメッセージを次のプロセスに渡すことで、もし、目的のプロセスまでマーキングされれば、最初のプロセスから目的のプロセスまでが接続されたと考えられる。

定義 4 マーキング

P を πF プロセス、 m をマーキング名とする。マーキング $m \triangleright P$ は次のように定義される。

$$m \triangleright P = \begin{cases} 0_m & \text{if } P = 0 \\ \bar{a}[b, m]. (m \triangleright Q) & \text{if } P = \bar{a}[b]. Q \\ a(\bar{b}). (m \triangleright Q) & \text{if } P = a(\bar{b}). Q \\ (m \triangleright Q) \mid (m \triangleright R) & \text{if } P = Q \mid R \\ (m \triangleright Q) + (m \triangleright R) & \text{if } P = Q + R \\ \nu \bar{b}(m \triangleright Q) & \text{if } P = \nu \bar{b}Q \end{cases}$$

また、マーキングされたプロセスのコミュニケーションは導出規則 MARK で補われる。

MARK:

$$\frac{\# \bar{c} = \# \bar{v}}{\bar{a}[\bar{c}, m]. P + P' \mid \{ \sum_i a(\bar{v}). Q_i + Q' \} \mid'}$$

$$\frac{}{\bar{a}[\bar{c}, m]. P + P' \mid \{ \sum_i a(\bar{v}, x). (x \triangleright Q_i + Q') \} \mid'}$$

' \mid' ' は MARK による導出を示す。 □

マーキングを用いて、2 つのプロセスの間の接続性が次のように定義できる。

定義 5 プロセスの接続性

今、プロセス P_1, \dots, P_m と環境 \mathcal{A} が与えられているとする。 $Sys = \prod_{k=1}^n \{P_k\}l_{k0}$ に対し、 $P_i \text{ con } P_j$ であるとは、次式が成り立つことを言う。

$$\prod_{1 \leq k \leq n, k \neq i} \{P_k\}l_{k0} \mid \{m \triangleright P_i\}l_{i0} \quad ((\rightarrow)^* \rightarrow)^*$$

$$\prod_{1 \leq k \leq n, k \neq i, j} \{P_k\}l'_k \mid \{m \triangleright P'_i\}l'_i \mid \{m \triangleright P'_j\}l'_j$$

ただし、 P'_k は P_k から導出されるプロセスである。また、ここでは導出 \rightarrow のアクション名は省略している。 □

マーキングを例 1 のレイバー L_1 に適用してみると、以下のように P から Q, R とマークされていくのが分かる。

$$\begin{aligned} &\{m \triangleright (\overline{call}[no]. 0 + \overline{break}. 0)\}1 \\ &\mid \{call(x). \bar{x}. 0 + \overline{break}. 0\}2 \mid \{R\}4 \\ = &\{\overline{call}[no, m]. 0_m + \overline{break}. 0\}1 \\ &\mid \{call(x). \bar{x}. 0 + \overline{break}. 0\}2 \mid \{R\}4 \\ \rightarrow &\{\overline{call}[no, m]. 0_m + \overline{break}. 0\}1 \\ &\mid \{call(x, y). \bar{x}[y]. 0_y + \overline{break}. 0\}2 \mid \{R\}4 \\ \rightarrow &\{0_m\}1 \mid \{\bar{no}[m]. 0_m\}2 \mid \{R\}4 \\ \rightarrow &\{0_m\}1 \mid \{\bar{no}[m]. 0_m\}2 \mid \{no. 0 + \overline{break}. 0\}3 \\ \rightarrow &\{0_m\}1 \mid \{\bar{no}[m]. 0_m\}2 \mid \{no(z). 0_z + \overline{break}. 0\}3 \\ \rightarrow &\{0_m\}1 \mid \{0_m\}2 \mid \{0_m\}3 \end{aligned}$$

従って、 L_1 に対して $P \text{ con } Q$ である。

一方、例 2 では、環境が \mathcal{A}_2 のように与えられていると、以下のように R までマーキングされずに中断されてしまう。

$$\begin{aligned} &\{m \triangleright (\overline{call}[no]. 0 + \overline{break}. 0)\}1 \\ &\mid \{call(x). \bar{x}. 0 + \overline{break}. 0\}2 \mid \{R\}4 \\ = &\{\overline{call}[no, m]. 0_m + \overline{break}. 0\}1 \\ &\mid \{call(x). \bar{x}. 0 + \overline{break}. 0\}2 \mid \{R\}4 \\ \rightarrow &\{\overline{call}[no, m]. 0_m + \overline{break}. 0\}1 \\ &\mid \{call(x, y). \bar{x}[y]. 0_y + \overline{break}. 0\}2 \mid \{R\}4 \\ \rightarrow &\{0_m\}1 \mid \{\bar{no}[m]. 0_m\}2 \mid \{R\}4 \end{aligned}$$

4 接続性と到達可能性

前節で示した接続性の定義5はシミュレーションの点から見た最も素朴なものである。本節では、接続性は以下で定義するプロセスの「到達可能性 (reachability)」と重要な関係があることを示す。到達可能性とは、プロセスがある場から別の場まで移動できるかどうかを言う。到達可能性および、場と場間のコミュニケーションに関する距離を導入することで、接続性を判定できる。以下の議論では、プロセスの組 P_1, \dots, P_n , および環境 $A = \langle \langle Loc, RL \rangle MV \rangle$, が与えられているとする。

定義6 到達可能性

プロセス P , 位置 l, l' に対し, $(P, l, A) \in MV$ かつ $l' \in A$ であるとき, $P : l \mapsto l'$ と書く。 A において, プロセス P が位置 l から移動できる範囲を $Reach(P, l)$ と書き, P の l からの到達範囲と呼ぶ。すなわち,

$$Reach(P, l) = \{m \mid P : l(\mapsto)^* m\} \quad \square$$

次に, 位置と位置間の距離を定義する。これは, 2個の位置の間の道の数を元にしており, 各々の位置にあるプロセスが最低いくつのコミュニケーションでデータが伝達されるかを表す。

定義7 位置間の距離

位置 l と l' に対し, 道の系列 $(l, l_2), (l_2, l_3), \dots, (l_{n-1}, l')$ $\in RL$ (これを, $\langle l, l_2, \dots, l_{n-1}, l' \rangle$ と書く。) がいくつか存在するものとする。この時, 最小の n を l から l' への距離という。すなわち,

$$d(l \rightarrow l') = n.$$

もし, そのような系列が存在しなければ, $d(l \rightarrow l') = \infty$ とする。また, 位置の集合 A, B に対し, A から B までの距離は以下のように定義される。

$$d(A \rightarrow B) = \min\{d(l \rightarrow l') \mid l \in A, l' \in B\}. \quad \square$$

例3 上述の A_1 における到達範囲と距離を図2に示す。この場合, L_1 に対して $P \text{ con } R$ が成り立っているが, メッセージが伝達される順 $P \rightarrow Q \rightarrow R$ に各々のプロセスの到達範囲間の距離を調べると,

$$d(Reach(P, 1) \rightarrow Reach(Q, 2)) = 1$$

$$d(Reach(Q, 2) \rightarrow Reach(R, 3)) = 1$$

が成り立っている。

一方, A_2 では $P \text{ con } R$ が成り立たなかったが, 上と同様に各プロセスの到達範囲 (図3) を調べると,

$$d(Reach(P, 1) \rightarrow Reach(Q, 2)) = 1$$

であるのに対し,

$$d(Reach(Q, 2) \rightarrow Reach(R, 3)) = \infty$$

である。

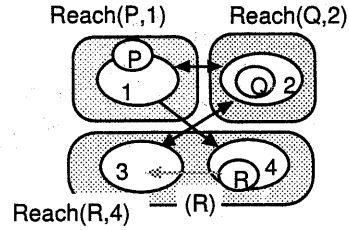


図2: 環境 A_1 における到達範囲

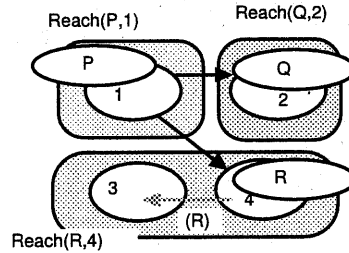


図3: 環境 A_2 における到達範囲

よって, 次の定理が成り立つ。

定理1 $Procs = \prod_{k=1}^n P_k$ に対して, $P_1 \text{ con } P_n$ であると

する。 $Sys = \prod_{k=1}^n \{P_k\}l_{k0}$ において $P_1 \text{ con } P_n$ であるためには, $1 \leq k \leq n-1$ について,

$$d(Reach(P_k, l_{k0}) \rightarrow Reach(P_{k+1}, l_{k+1})) = 1$$

が成り立つ。 \square

定理1の証明は [8] に譲る。

5 環境の補償

環境の損傷によりシステムが正常に動作しない場合, 例えば, 移动通信で建物の陰に入って電波が届き難くなると通信が途絶えることがあるが, それは陰になっている場所に電波が届くよう基地局を新設して補償できる場合がある。これは, πF のモデルでは, 位置を増やしたり道を付加することに相当する。本節では, 定理1に基づき, 環境の補償を場のモデルの上で行う枠組みを示す。

今, $A = \langle \langle Loc, RL \rangle MV \rangle$, および, プロセスの組 P_1, \dots, P_n が与えられているとする。ここでは, プロセス

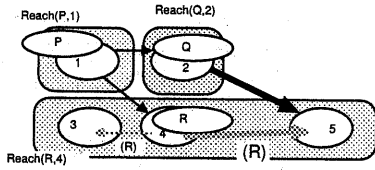


図 4: 環境 A_2 に対する補償

の内の P_k と P_{k+1} の接続性が満足されていない状況にあるとし、その補償を考える。

補償には、3通り考えられる。物理的な補償が簡単な順に並べると次のようになる。補償後の量には'をつけている。

- (I) 移動の拡張 ($MV' \supset MV$)
 - (II) 道の拡張 ($RL' \supset RL$)
 - (III) 位置の拡張 ($Loc' \supset Loc$)
- (III) の場合、必然的に移動または道の拡張が行われる。

例 4 A_2 では P は R と接続できない。例 3 のように、 $Reach(Q, 2)$ と $Reach(R, 4)$ の間で定理 1 の条件が満足されないからである。今、位置 2' と 3' の間に山などの障害物があり、どうしても通進路が開けない場合、新たに位置 5' を用意することで P から R への接続性を補償できる (図 4)。これは、移动通信の基地局を新たに設営することなどに相当する。

6 まとめ

本研究では、モバイルシステムを環境の側面から管理するため、形式記述技法を用いる方法について考察した。システムの運用という点から考えた時、モバイルシステムの通信プロトコルは必ずしも有線ネットワークと同様に上位層と下位層とを区別できるわけではないというのが、本研究の主張である。並行計算 πF は通信システムを上位層、下位層両面で支援するために考えられた形式記述技法である。本稿では、上位層で接続性が確保されているにも関わらず、下位層での不具合によって通信が妨げられる例に注目し、下位層の性質を明確にすることを試みた。モデル上での補償は単にグラフのつながり替えに過ぎないが、これにより現実の補償作業を示唆できるのではないと思われる。本稿では、紙数のため補償アルゴリズムを示すまでに至らなかったが、そのアルゴリズムは [8] で報告する。

プロセス記述から位置情報を分離したことで、プロセス動作記述が簡潔になった。一方で、位置の膨大な情報を列挙するのは一般に困難であるので、このような地理情報はデータベースで管理し、その下でプロセスの動作をシミュレーションによって調べることが現実的な利用法だと思われる。その意味で、 πF はシミュレーション指向とも言える。プロセスの動作が環境に及ぼす影響の調査と仕様記述支援法を研究するため、現在シミュレータを開発中である。

参考文献

- [1] G. Berry and G. Boudol, The chemical abstract machine, *Theoretical Computer Science*, vol.96, pp.217-248, 1992.
- [2] R. Milner, J. Parrow and D. Walker, A calculus of mobile processes, Part I and II, *Journal of Information and Computation*, vol.100, pp. 1-77, September, 1992.
- [3] D. Sangiorgi, Locality and non-interleaving semantics in calculi for mobile processes, Technical Report ECS-LFCS-94-282, Laboratory for Foundations of Computer Science, Department of Computer Science, University of Edinburgh, UK, 1994.
- [4] R.M. Amadio, and S. Prasad, Localities and failures, *Proceedings FST-TCS'94, LNCS 880*, pp. 205-216. Springer-Verlag, 1994.
- [5] Y. Isobe, Y. Yutaka and K. Ohmaki, Approximative Analysis by process algebra with graded spatial actions, *Proceedings AMAST'96, LNCS 1101*, pp.336-350, Springer-Verlag, 1996.
- [6] T. Ando, K. Takahashi and Y. Kato, On a Concurrency calculus for design of mobile telecommunication systems, *Proceedings FORTE(X)/PSTV(XVII)'97*, pp.535-546, 1997.
- [7] 安藤敏彦, 高橋薫, 加藤靖, πF を用いた車々間プロトコルの形式記述, 情報処理学会研究報告, Vol.99, ITS-2, pp.69-76, 1999.
- [8] T. Ando, K. Takahashi, Y. Kato and N. Shiratori, Maintenance of Ambiences for Mobile Systems, *Proceedings ICPP-99, 1999 (to be appeared)*.