

移動体環境におけるアクティブデータベースの安全性について

村瀬 亨† 塚本昌彦‡ 西尾章治郎‡
†住友電気工業(株) システムエレクトロニクス研究開発センター
〒554 大阪市此花区島屋1-1-3
E-mail: murase@rinfo.sumiden.co.jp

‡大阪大学工学部 情報システム工学科
〒565 吹田市山田丘2-1
E-mail: {tuka,nishio}@ise.eng.osaka-u.ac.jp

筆者らは、無線通信による移動体計算環境におけるデータ統合を実現するうえで、移動や省電源切断動作などをイベントとしてとらえ、アクティブデータベースにおけるECAルールを適用することが有効であることをこれまで示してきた。しかし、ECAルールはシステム間で無限ループに陥らないかどうかの判定が必要であり、さらに移動体に適用する場合は複数システム間の整合性の問題や移動によって、ルールの実行シーケンスが停止するかどうかの判定が困難である。本稿では、このようなシステム動作の安全性について、特にルールの停止性について、移動による影響を含めて判定するための方法について考察する。そのなかで、イベント発生ホストとトリガー連鎖の関係をマージグラフと呼ぶグラフとして表現することで効率よく安全性が判定できることを示す。

On Safety of Active Databases in Mobile Computing Environments

Toru MURASE † Masahiko TSUKAMOTO ‡ Shojiro NISHIO ‡
† Systems and Electronics R & D Center, Sumitomo Electric Industries, Ltd.
1-1-3, Shimaya, Konohana-Ku, Osaka 554, Japan
E-mail: murase@rinfo.sumiden.co.jp

‡ Dept. of Information Systems Engineering, Faculty of Engineering, Osaka University
2-1, Yamadaoka, Suita 565, Japan
E-mail: {otuka,nishio}@ise.eng.osaka-u.ac.jp

For realizing the data integration in mobile computing environments, we showed that the notion of ECA rule in active databases is effective to describe events happening in these environments such as moving of computers, disconnected operation for energy-saving, and so on. Though the ECA rule is useful, it has inevitable difficulty for guaranteeing that the rules can work without falling into infinite loop. Furthermore, if the system is applied for mobile computing environments, the complexity for predicting termination will be increased due to the facts that (1) trigger-chaining spreads over multiple computer systems, and (2) each computer system has the mobile capability. In this paper, we show analysis methods for the safety of our assuming active database. Especially, a method how to check the termination of ECA rules is proposed, where we introduce E-merge graph indicating the relationship among system events and trigger-chaining.

1.はじめに

無線通信機能を備えたパーソナルコンピュータ (PC) や携帯情報端末 (PDA) による移動体計算環境が実用化の段階に入りつつある [3] [4].

筆者らはこれまでに移動体計算モデルにおけるデータ統合の有効性について議論し [8] [9] [11] [12] [13] [14], 特にアクティブデータベース [5] [13] の利用の可能性について述べた [10]. 移動体計算環境においては, すべての計算機は固定ホストと移動ホストとからなっている. 固定ホストとは, 移動体計算環境において, ネットワーク・トポロジー上その位置を変えない計算機をさす. これに対して, その位置を変更し, ネットワーク・トポロジー上の位置を変更する計算機を移動ホストと呼ぶ. この移動体計算環境のなかで, システム内で起こる移動計算機の通信セル間の移動や移動計算機 (移動ホスト) 自身の省電力設計のためにとられる移動体ホストサーバとの間の切断動作 (Disconnected Operation) [7] などをすべてイベントとしてとらえ, それらに対してアクティブデータベースにおけるECAルールを適用する AMDS (Active Mobile Database System) として提案した. このことで AMDS を移動体計算環境での異機種分散システムの共通基盤として提供しうることが述べた.

AMDS は, 移動体計算環境におけるさまざまなイベントを直接データベースのデータ更新の実行に結びつけたり, あるいはトリガーやアラートとして, 他のノードへの通知やシステム外部への出力に利用できるという利点を有している.

AMDS では, 移動体計算環境における各ノード毎に ECA ルールの実行機能が配置されている [10]. 各ノードには目的に沿った実行ルールが, イベント (Event), 条件 (Condition), アクション (Action) の形態で格納されている. これらのルールは, イベントの発生に伴い, 条件が合致すればアクションの実行を行う. ひとつのアクションの論理的な粒度は, これまでの実装評価によると多くの内容を記述として押し込めるよりもルー

ルの数がふえても単純な内容にしたほうが応用度が高いという傾向を示している [10]. その結果, 多くのルールではイベントの発生したホストでユーザの目的に合致したアクションを実行するよりも, むしろ他のノードに対してそのイベントを通知し, ホスト間で目的に沿って予め配置されているルールが連鎖的に実行される.

いっぽう, AMDS においては, 各ノードに配置された ECA ルールが, 相互にどのようなシーケンスで実行されるのか, あるいはそれぞれのルールが相互にどのように振る舞い, かつ停止するかの保証をすることが必要となる. 特に, 特定のユーザ目的に沿って作成配置されたルールであっても, 移動ホストの移動の仕方によっては, 目的に合致した動作をしないばかりか, ルールの実行自体がホスト間で無限ループに陥ってしまい, AMDS として停止しない恐れがあるからである.

このような点に関し, 従来からアクティブデータベースの分野では, ルールの停止条件を含めた安全性について, その動作特性を次のように定義している [1].

- ①停止性 (Termination) : アクティブデータベースに対するあらゆる状態変更が完了した後, ルールの処理が停止することが保証されていること.
- ②合流性 (Confluence) : 実行権に優先付けがなされていないルール間で, 実行順序によってデータベースの最終状態に差異が生じうるかどうか. 言い換えれば, ルールの処理の過程で複数のルールが同時に発火した場合, ルール実行停止時のアクティブデータベースの最終状態がルールの実行順序に依存しないこと.
- ③可観測決定性 (Observable Determinism) : ルールのアクションがアクティブデータベースの外部から見えることを観測性と呼ぶ. さらに, 実行権に優先付けがなされていないルール間で, 観測性のあるアクションの順序や結果がルールの実行順序に依存

しないこと。

これらの特性分析は、単一のアクティブデータベース内のルール相互の安全性について解析手法に適用されたものである [2] [6] [15]。

移動体計算環境においては、ネットワーク内の複数の計算機上にアクティブデータベースが複数存在しており、相互のルールの実行関係を検討する必要がある。さらに、ルールを保持している移動ホストと固定ホストとの接続関係が移動により変化することによるトポロジーの変化を解析に加える必要がある。

本稿では、移動体計算環境におけるアクティブデータベースの安全性のなかでも特に移動によるルール配置の変化を考慮したルールの停止性について述べる。他の安全性の特性である合流性や可観測決定性については、別の機会に停止性と同様にして解析する。

以下、第二章では、AMDSのECAルールの実例をあげ、配置により停止しないケースを述べる。第三章では、本稿での議論の中心となる移動ホストの位置関係の縮退とルール相互の安全性に関する解析を展開する。第四章では、これらの解析の適用を分析して今後の課題を明らかにし、第五章でまとめとする。

2.AMDSでのECAルールの実例

本章では、AMDSでのECAルールの配置例を示し、停止性が問題になる場合を具体的に述べる。

AMDSでのルール実装例を図1の遊園地入場者端末システムに示す。入場者はそれぞれ入場者端末（移動ホスト） m_1 、 m_2 、 m_3 を携帯している。施設側では、入場者が利用する施設にそれぞれチェックイン装置（固定ホスト） v_1 、 v_2 を設置している。チェックイン装置は、入場者端末が施設に入場したことを検知する機能を持っている。チェックイン装置 v_1 には2つのルールが配置されている。ルール1は、チェックイン装置1に入場者端末がチェックインしたこと

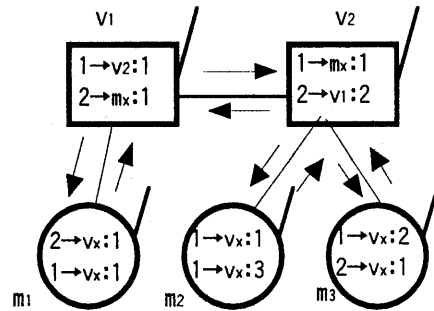


図1.AMDSのルール実装例

をイベント1として、

ルール1:1 → V_2 :1

Event: 入場者端末がチェックイン
(イベント1)

Action: イベント1をチェックイン装置 v_2 に伝える。

ルール2:2 → m_x :1

Event: 入場者端末 m_x がチェックイン
(イベント2)

Action: イベント2をチェックイン装置 v_1 施設内にいるすべての入場者端末に伝える。

ここで、条件に適合するホストをさす場合、 m_x と表す。同様に、チェックイン装置 v_2 、入場者端末 m_1 、 m_2 、 m_3 にはそれぞれのルールが配置されている。ルールにおいて、 v_x は、入場者端末が直接接続されている固定ホストを表す。ここで、無線リンクにより、チェックイン装置1に入場者端末1が接続され、チェックイン装置2に端末2、3が接続されたと仮定する。固定ホスト v_1 において、ルール1の発火を契機とするルール起動関係の流れを矢印で示す。ルール1の発火により、少なくともチェックイン装置 v_1 、チェックイン装置 v_2 、入場者端末 m_3 、チェックイン装置 v_2 、チェックイン装置 v_1 、入場者端末 m_1 、入場者端末 v_1 の流れによって無限ループが形成され、ルール実行が停止することはない。

このように、それぞれのホストに配置された個別のルールは妥当であっても、複数のホスト間では、ルール設計者の意図に反し、無限ループを形成してしまう場合が存在する。移動体計算環境では、移動ホストと固定ホストとの位置関係の変化がルールの実行シーケンスに影響を与えるため、停止性を含めた安全性の解析手法が必要となる。

3.安全性解析手法

本章では、まずホストとルールの配置関係についてのモデルを提示する。続いて、解析手法としてトリガグラフ、さらにそれを拡張した σ マージトリガグラフについて述べる。

3.1.要素の定義

ホストの集合を以下のように定義する。

V : 固定ホストの集合

M : 移動ホストの集合

また、 E をイベントの集合とする。

固定ホスト $v \in V$ は、相互に有線通信リンクで接続されている。移動ホスト $m \in M$ は、移動によって固定ホスト v との接続関係が変化する。

固定ホスト v と、そこに無線通信リンクによって接続されている移動ホスト m との関係、すなわち移動ホストの位置割当 l を次のように定義する。

$l: M \rightarrow V$: 移動ホストの位置割当

ここで、移動ホストは単一の固定ホストに割当されると仮定する。また、複数の移動ホストが一つの固定ホストに接続されることもありうることを仮定する。

例えば、図1において、円は移動ホスト、四角は固定ホストを表すが、図に示されるような位置関係にあるホストの関係はそれぞれ、

$$l(m) = v_1$$

$$l(m) = v_1$$

$$l(m) = v_2$$

となる。

ネットワーク全体の位置割当の全体からなる集合を Loc とおく。

次に、各ホストにおけるECAルールを次のように定義する。

(1) 固定ホストに対するECAルール:

$$r: E \rightarrow 2^{(V \cup \{O\}) \times E}$$

O は、無線リンクを介して接続される移動ホストの全体を表す。

$$r(e) = \{(x_0, e_0), \dots, (x_n, e_n)\} \\ (e \in E, x_i \in V \cup \{O\}, e_i \in E (i=0, \dots, n))$$

は、直観的には、イベント e が起これば、ノード v_i に対してイベント e_i を送信するというルールを表している。また、このような r の全体を R_V と記す。

(2) 移動ホストに対するECAルール:

$$r': E \rightarrow 2^{(I, O) \times E}$$

I はルールにおけるアクションが移動ホスト内部のローカルな動作であることを表す。このような r' の全体を R_m と記す。

図1の例では、 v_1 のECAルールは、

$$r(e_1) = \{(v_2, e_1)\}$$

$$r(e_2) = \{(O, e_1)\}$$

$$r(e_3) = \emptyset$$

本モデルでは、固定ホストから移動ホストを選別してイベントを送信することはできないと仮定している。実際上の選別送信は、受信した移動ホスト側で選択的に受信することでこれと等価な通信が行えるので、本モデルで取り扱うことができる。

ECAルールの割当を次のように定義する。

$$a_v: V \rightarrow 2^R$$

$$a_m: M \rightarrow 2^{R_m}$$

ここで、

A_v : a_v の全体集合

A_m : a_m の全体集合

とおく。

自発的イベントとは、 $(V \cup M) \times E$ の部分集合のことをいう。ここで、自発的イベント集合、

$$G = \{(x_0, e_0), \dots, (x_m, e_m)\}$$

$$x_0, \dots, x_m \in V \cup M, e_0, \dots, e_m \in E$$

は、ノード x_i においてイベント e_i が自発的に

起こりうることを表す。例えば、図1の例では、

$$G = \{(v_1, e_1), (v_1, e_2), (v_2, e_1), (v_2, e_2)\}$$

となる。

3-2. 実行モデル

本節では、ECA ルールの実行モデルを定義する。

$l \in Loc$, $a_v \in A_v$, $a_m \in A_m$ とする。 $x, x' \in V \cup M$, $e, e' \in E$ に対して、次のいずれかの条件を満たす場合、 x が x' を誘発するといひ、 $(x, e) \rightarrow (x', e')$ と記す。

$$1. x \in V, x' \in V, \exists r: r \in a_v(x), (x, e') \in r(e)$$

$$2. x \in V, x' \in M, r \in a_v(x), (O, e') \in r(e),$$

$$l(x') = x$$

$$3. x = x' \in M, \exists r: r \in a_m(x), (l, e') \in r(e)$$

$$4. x \in M, x' \in V, \exists r: r \in a_m(x), (O, e'),$$

$$l(x) = x'$$

直観的には、ノード x においてイベントが発生し、その結果、ノード x' において新たなイベント x' が発生することを意味する。つまり、ルールが固定ホストに割当られ、かつ固定ホストに対するルールである場合か、ルールが移動ホストに割当られたルールでそれが誘発すべきホストがその移動ホストと直接リンクをもつ固定ホストであるような関係のルールである。ノード、イベント組の集合の列 $E(i)$ ($i=0, \dots, n$ あるいは $i=0, \dots$) が活性化列であるとは、各 i に対して次のいずれかが成立することをいう。

$$(1) (x_i, e) \in E(i)$$

$W = \{(x'_1, e'_1), \dots, (x'_m, e'_m)\}$ の各元を誘発し、 $E(i+1) = [E(i) \cdot (x, e)] \cup W$ がなりたつ。

(2) $E(i+1) = E(i) \cup W$, ただし $W \subseteq G$ (1) は、第 i ステップから次のステップに移った時に、その誘発されたイベントがまだ残っていることを示す。(2)により、次のイベントの集合は、自発的イベントの集合からなっていることを示す。非活性化列とは、(1)の条件のみが成立するような活性化列のことをいう。

ここで、ノード、イベント組の集合 E_0 が安全であるとは E_0 ではじまる任意の非活性化列が有

限列であることをいう。また、位置割当 l が安全であるとは空集合 $E_0 = \emptyset$ ではじまる任意の有限活性化列 E_0, \dots, E_p において E_p が安全であることをいう。そして、システムが安全であるとは、任意の位置割当 l に対して l が安全であることをいう。

本モデルにおいて安全なシステムは、以下に述べる誘発有限性という好ましい性質が成立することを示せる。ノード、イベント組の列 (x_i, e_i) ($i=0, \dots, n$ あるいは $i=0, \dots$) が誘発列であるとは、任意の i に対し、 (x_i, e_i) が (x_{i+1}, e_{i+1}) を誘発することをいう。

[補題1(誘発有限性)] 位置割当 l に対し、自発的イベントから始まる任意の誘発列が有限であること、 l が安全であることは同値である。[証明]: 誘発列が有限の場合に、誘発列の長さに関する帰納法を用いて示すことができる。[証明終]

3.3. トリガーグラフ

ここで、ルールにおけるトリガーの関係を表したトリガーグラフを考える。トリガーグラフは、イベントの発生するホストをノードとし、イベントの誘発関係をアークとする有効グラフである。すなわち、

$$\text{ノード: } (U \cup M) \times E$$

$$\text{アーク: } (x, e), (y, e') \text{ ただし,}$$

$$1. \exists r(e) \in a_v(x): (y, e') \in r(e) \text{ あるいは,}$$

$$2. \exists r(e) \in a_v(x): (O, e') \in r(e) \text{ かつ } l(y) = x \text{ あるいは,}$$

$$3. \exists r(e) \in a_m(x): (O, e') \in r(e) \text{ かつ } l(x) = y$$

このトリガーグラフを用いて、移動ホストの固定ホストとの接続関係が変化することを考慮してルールの停止性を調べることができる。

図2にルールの配置とそのトリガーグラフを示す。固定ホスト a では、イベント1が発生すると、 b に対してアクション2を送り、イベン

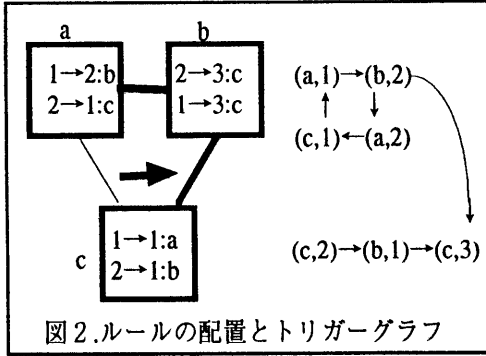


図2. ルールの配置とトリガーグラフ

ト2が発生すると移動ホストcに対してアクション1を送るルールを割当てている。同じように固定ホストbと移動ホストcにもそれぞれルールが割当てられている。このルールの発火の誘発関係のトリガーグラフは右図のようになる。ここで、システムが安全であるためには、次の条件を満たす必要がある。

[定理1]: 任意の位置割当てに対してトリガーグラフに自発的イベントから可到達なループがないこととシステムが安全であることは同値である。

[証明]: トリガーグラフにループがあれば、そのループを用いて無限の誘発列が存在する場合、同じノードが二度以上出現していることになり、その区間がグラフ上ループであることが示せる。以上のことにより、補題1を用いて、定理が証明される。[証明終]

これにより、トリガーグラフを構成し、可到達なループの検出をすることにより、システムの安全性が確かめられる見通しが立った。しかし、対象となるすべての移動ホストと移動ホストサーバとの関係、すなわち移動ホストのすべての位置割当てについてを調べるため、トリガーグラフの構成回数は $|V|^M$ 回になる。例えば、固定ホスト2台、移動ホスト2台の場合の位置割当ては図3のようなパターンを取り得る。システムが停止性において安全であるには、少なくとも移動ホストからみて、接続先の固定ホストに関係なく無限ループが存在してはならない。そこで、先にトリガーグラフにおいて調べ上げた場合のうち、単に移動ホ

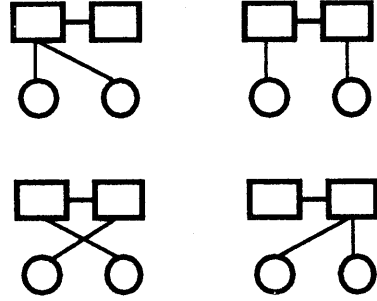


図3. 固定ホスト2台×移動ホスト2台の位置割当て

ストが接続される固定ホストが区別されたために発生した割当てを統合するほうが有利である。そこで、この数を減らし、簡便に調べる方法が必要となる。

3.4. σ マージトリガーグラフ

トリガーグラフでは、ノードにホストをとり、アークとしてECAルールの誘発関係を取って構成している。

トリガーグラフの構成の回数を減らすために、別のグラフの構成方法を導入する。 V の分割 p とは、 $U_p = V, \forall x, y \in p: x \cap y = \emptyset$ が成り立つような V の部分集合の集合のことをいう。 V の分割全体を P と記す。拡張位置割当て σ とは、 $\sigma: M \rightarrow P$ のことをいう。分割 p がすべて単一元からなるとき、拡張位置割当ては前述の位置割当てと同様となる。拡張位置割当ての全体を LOC と呼ぶことにする。任意の x に対して $\sigma(x)$ が単一元の集合となるようなもの全体が Loc となる。拡張位置割当て σ に対し、 σ マージグラフは次のようなグラフのことをいう。

ノード: $(V \cup M) \times E$

アーク: $(x, e), (y, e')$ ただし、

1. $\exists r(e) \in a_r(x): (y, e') \in r(e)$ あるいは、
2. $\exists r(e) \in a_r(x): (O, e') \in r(e)$ かつ $x \in \sigma(y)$ あるいは
3. $\exists r(e) \in a_m(x): (O, e') \in r(e)$ かつ $y \in \sigma(x)$

ここで、有効ループを移動体の異なる位置を参照していないようなループと定義する。

〔定理2〕：分割 n に対して、システムが安全であることと、 n に対する任意の拡張位置割当 σ に対して σ マージグラフに自発的イベントから到達可能な有効ループがないこととは同値である。

〔証明〕：定理1を用いて、有効ループがあれば安全でないことは容易に示せる。安全でなければ、有効ループがあることは、無限の非活性化列がないことから示される。〔証明終〕

固定ホスト v, v' が互いに素であるとは、トリガーグラフで v を含むノードがすべて非連結であることをいう。固定ホスト v, v' のマージトリガーグラフとは、

$\sigma : \forall m, \sigma(m) = \{x \mid x \neq v \text{ かつ } x \neq v'\} \cup \{v, v'\}$ を用いた σ マージトリガーグラフのことをいう。

〔定理3〕固定ホスト v, v' が互いに素なら、システムが安全であることと、 v, v' の σ マージトリガーグラフにループがないことと同値である。

〔証明〕：安全ならばトリガーグラフにループはない。安全かつ v, v' が互いに素であるならば、 v, v' のマージトリガーグラフに新たなループができることはない。〔証明終〕

拡張位置割当 σ では、直観的には、無限ループの形成が最低2本のアークが存在するはずとの前提にたっており、 σ マージトリガーグラフの導入により、ひとつでも有効ループを形成するトリガーグラフが検出されれば、そのECAルールの位置割当は安全でないことがいえるため、停止性の検証が効率的に実行できる見通しがたった。

特に、 σ として $V \times V$ を選ぶと、 $V \times V$ トリガーグラフの有効ループを調べることでシステムの停止性を確かめられることになる。トリガーグラフでの構成回数は増えるが、グラフのアークが $|M| \times |V|$ のオーダーで増加するので、アルゴリズムの複雑さは変わらない。

4. 解析手法の評価

本章では、 σ マージトリガーグラフによる解析手法の評価をおこなう。

本稿では、図2を例にとり、移動ホストの位

置割当によって形成されるトリガーグラフの構成回数を減らすために拡張位置割当を導入した。そして、有効ループがあれば安全でないことを証明し、 σ マージトリガーグラフの導入でひとつでも有効ループが検出されれば、そのECAルールが移動ホストの位置割当によって停止しないことがありうることを述べた。これにより、移動によるECAルールの安全性についての検証が効率的にできるようになった。

停止性の解析手法としては、まだ未解決の問題も残っている。今回、ルールの位置割当については、移動という動作後の場合で論じた。正確には、移動というイベントが反映されない場合も出てくると考えられる。

本稿では、ECAルールのコンディション（条件）の部分を検討しなかったため、安全システムの誘発有限列というシステム解析上好ましい性質が得られたが、一般の場合には解析は大幅に複雑になる。今後は、条件部を検討した効率よい解析手法を考える必要がある。

5. まとめ

本稿では移動体計算環境におけるデータ統合のためのAMDSにおけるECAルールの安全性について議論した。 σ マージトリガーグラフの導入により、安全性のうち、停止性について効率よく判定できることを示した。今後、4章で論じた他の特性についても検証する方法を検討していく。

謝辞

末筆ながら、アクティブデータベースAMDSの基本概念構築にあたり、大阪大学工学部情報システム工学科西尾研究室の方々には、議論を通して有益なコメントをいただいた。また、本研究の一部は住友電気工業（株）からの研究補助費によるものである。ここに記して謝意を表す。

参考文献

[1] Aiken, A., Hellerstein, J., Widom, J.:

- Static Analysis Techniques for Predicting the Behavior of Active Database Rules, ACM Transaction on Database systems, Vol. 20, No. 1, pp. 3-41, March 1995.
- [2] Hanson, E.: An Initial Report on The Design of Ariel: A DBMS With an Integrated Production Rule Systems, SIGMOD RECORD, Vol. 18, No. 3, Sept. 1989.
- [3] Imielinski, T., and Bardrinath, B. R. : Mobile Wireless Computing: Solutions and Challenges in Data Management, Technical Report DCS-296, Dept. of Computer Science, Rutgers University, 1993.
- [4] Imielinski, T., and Badrinath, B. R. : Mobile Wireless Computing, Communications of the ACM, Vol. 37, No. 10, pp. 18-28, Oct. 1994.
- [5] 石川博: アクティブデータベース, 情報処理, Vol. 35, No. 2, pp. 120-129, Feb. 1994.
- [6] Kifer, M., Ramajrishnan, R., Siberschatz, A.: An Axiomatic Approach to Deciding Query Safety in Deductive Database, Proc. of PODS'88, pp. 52-60, 1988.
- [7] Kistler, J. J., and Satanarayanan, M.: Disconnected Operation in the Coda File System, Proc. of the 13th Symposium on Operating Systems Principles, pp. 13-16, 1991.
- [8] 劉渤江, 塚本昌彦, 西尾章治郎: 移動体計算環境におけるデータベースビュー, Proc. of Advanced Database System Symposium'94, pp. 9-18, 1994.
- [9] 劉渤江, 仲秋朗, 塚本昌彦, 西尾章治郎: 移動体計算環境におけるデータベースビュー定義言語, 電子情報通信学会技術研究報告, Vol. 95, No. 81 (DE95-1~8), pp. 25-32, May 1995.
- [10] 村瀬亨, 塚本昌彦, 西尾章治郎: アクティブデータベースシステムによる移動体計算環境におけるデータ統合, 電子情報通信学会データ工学研究会技術研究報告, Vol. 95, No. 287, pp. 41-48, Oct. 1995.
- [11] 仲秋朗, 劉渤江, 塚本昌彦, 西尾章治郎: 移動体データベースのビュー維持手法, 情報処理学会研究報告, Vol. 95, No. 31 (95-DBS-102), pp. 33-40, 1995.
- [12] 白井博章, 仲秋朗, 劉渤江, 塚本昌彦, 西尾章治郎: 移動型データベースのためのビュー機構の設計および実装, 電子情報通信学会技術研究報告, Vol. 95, No. 148 (DE95-27~37), pp. 1-8, July 1995.
- [13] Stonebraker, M.: The Integration of Rule Systems and Database, IEEE Transactions on Knowledge and Data Engineering, Vol. 4, No. 5, pp. 415-423, Oct. 1992.
- [14] Tsukamoto, M., Liu, B., and Nishio, S.: MobiView: A Database View for Mobile Computing Environments, ISE-TR-95-013, Dept. of Information Systems Engineering, Faculty of Engineering, Osaka University, June 1995.
- [15] Vardi, M.: Decidability and Undecidability Results for Boundedness of Linear Recursive Queries, Proc. of PODS'88, pp. 341-351, 1988.