モバイル環境向けエージェント移動制御方式の適用効果

川上 憲治† 広重 一仁† 吉岡 信和‡ 本位田 真一‡＊

あらまし 近年，携帯電話やPDAなどの携帯端末が広く普及し，利用場所を選ばずに，様々なアプリケーションを利用することが可能となった．しかしながら，携帯端末はバッテリの持続時間や小型化のため，アプリケーションで利用できる計算リソースが限られるという制約がある．この制約を解決するための機構として，筆者らはモバイルエージェントを基盤とするミドルウェアについて提案してきた．本稿では，まず筆者らが提案したモバイルエージェントを基盤とするミドルウェアにおける移動制御方式について説明し，計算機シミュレーションによりその適用効果および移動制御に伴うオーバヘッドの影響を明らかにする．

キーワード 携帯端末，モバイルエージェント，移動制御

Effects of Agent Migration Control Methods for Mobile Environments

Kenji KAWAKAMI† Kazuhito HIROSHIGE† Nobukazu YOSHIOKA‡ and Shinichi HONIDEN‡＊

Abstract In recent years, mobile terminals such as cellular phones and PDAs have come into widespread use and various applications have been used anytime and anywhere. However, applications executed on the mobile terminals lack sufficient computational resources because of limited battery lifetime and terminal miniaturization. In order to overcome such a limitation, we have proposed the mobile agent-based middleware. In this paper, we describe our proposed agent migration control methods on the middleware and clarify the effects of these methods and the influences of overheads caused by the migration control through computer simulations.

Keyword Mobile Terminal, Mobile Agent, Migration Control

1. はじめに

近年，携帯電話やPDAなどの携帯端末が広く普及し，利用場所を選ばずに，様々なアプリケーションを利用することが可能となった．また，個人利用に限らず，企業の情報システムにおいても携帯端末を利用する事例が増えており，携帯端末の重要性は年々高まりつつある．

しかしながら，携帯端末はバッテリの持続時間や小型化のため，アプリケーションで利用できる計算リソース(CPU，メモリ，通信帯域幅など)が限られるという制約がある．これに対して，アプリケーションが他のコンピュータの豊富なリソースを利用でき，また，計算リソースや通信環境の変化にも適応できる機構があれば，ユーザは携帯端末の制約を意識することなくアプリケーションを利用できる．

筆者らはこのような機構を実現する手法として，モバイルエージェントを基盤とするミドルウェアについて検討してきた[1]-[4]．ここで，モバイルエージェントはミドルウェアが提供するライブラリとして動作し，他の端末に移動してそのリソースを利用し処理を実行する．

本稿では，まず筆者らが提案してきたモバイルエージェントを基盤とするミドルウェアにおける移動制御方式について説明する．そして，提案システムの適用効果を検証するための計算機シミュレーション結果を示す．具体的には，(ⅰ)提案システムによる処理性能の改善効果，(ⅱ)エージェント移動制御に伴うオーバヘッドの影響について述べる．

2. モバイル環境における課題

携帯端末では，バッテリの持続時間や小型化のため，CPUやメモリといったアプリケーションで利用できる計算リソースが制限される．このような携帯端末の制約をユーザに意識させないようにするためには，携帯端末上で実行されるアプリケーションは，他のコンピュータの豊

†日本テレコム株式会社, Japan Telecom
‡国立情報学研究所, National Institute of Informatics
＊東京大学, The University of Tokyo

富な計算リソースを柔軟に利用することにより，限られた処理能力を補うことが求められる．

また，携帯端末が利用できる計算リソースやバッテリ残量などは時間とともに変化し，さらに携帯端末が異なるネットワーク間を移動する場合，通信帯域幅や通信コストといったネットワーク環境が変化する．このような環境変化がアプリケーションの実行中に発生した場合に，変化に合わせて端末の処理を他の端末への移動も含めてダイナミックに変更することができれば，処理を効率的に実行できる．このような機構を実現するためには，まず環境の変化を検出し，そして最も効率的に処理を実行できる端末を自端末も含めて適応的に選択することが求められる．

3. 提案システムの構成

前章で述べた携帯端末の課題を解決するために，筆者らはモバイルエージェント(以下，MA と略す場合あり)を基盤とするミドルウェアについて検討してきた[1]-[4]．モバイルエージェントを利用することにより，ミドルウェアは上記の課題を柔軟に解決することができる．

図 1に提案システムの構成を示す．管理 MA 層は，マネージャとコーディネータで構成される．マネージャは，アプリケーションとモバイルエージェントもしくは異なるモバイルエージェント間のブローカとして機能する．コーディネータは，各端末のコーディネータと協調して動作し，サービス MA の移動先端末を決定するための交渉を行う．サービス MA 層は，複数のサービス MA と複数のモニタ MA で構成される．各サービス MA は，通信やデータ入出力など，アプリケーションが利用する個々の機能を提供する．各モニタ MA は，アプリケーションまたは他のモバイルエージェントが利用する計算リソースなどの変化を監視する．ランタイム層は，モバイルエージェントの実行を制御するための基本的な機能を提供する．

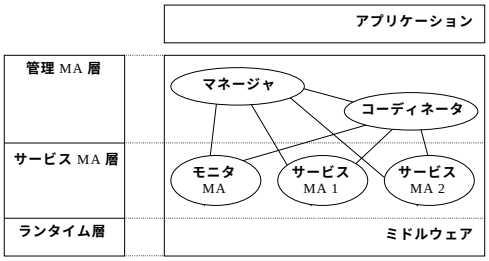

図 1 システム構成

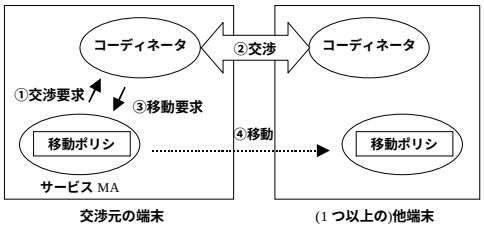

図 2 移動制御方式の概要

4. 移動制御方式

4.1. 概要

提案する移動制御方式は，図 2に示すようにコーディネータがサービス MA からの交渉要求(①)を受けた後，他端末のコーディネータと移動先端末を決定するための交渉を行う(②)．交渉終了後，コーディネータはサービス MA に対して移動要求を行う(③)．

各サービス MA には，交渉を開始するタイミングを規定するための"移動条件"と，交渉時に最適な移動先端末を判定するための"評価基準"が予め定義されており，ここでは，これらをサービス MA の"移動ポリシ"と呼ぶ．例えば，移動条件には，サービス MA が最低限必要とする CPU 能力やメモリ等のリソース割当量などを定義する．また，評価基準には，サービス MA を利用するアプリケーションの処理時間や通信コストを算出するための評価式などを定義する．

移動ポリシは各サービス MA の内部データとして保持される．また，サービス MA とともに移動ポリシも移動するため，移動ポリシを移動先の端末に配布する仕組みは必要としない．

4.2. 交渉開始のトリガ

提案方式では，あらかじめ登録された移動条件を満たす環境変化(例えばリソース割当量の変化等)がノード内に生じた場合に，それをトリガとして移動先決定のための交渉を開始することを基本とする(提案方式 1)[2]．さらに，リソース割当量の豊富な端末から他の端末に通知を行って交渉を開始するという仕組みを追加し，一層の効率化を図れる方法を提案した(提案方式 2)[3]．以下に，これら 2 つの方法を詳細に述べる．

提案方式 1：図 3に示すように，まずサービス MA はモニタ MA に移動条件を含む監視要求を通知する(①)．モニタ MA は監視要求に基づき，OS またはミドルウェア(ランタイム層)が提

供するリソースマネージャからリソースに関する情報を取得する(②)．サービス MA の移動条件が満たされると(③)，モニタ MA はコーディネータに交渉開始要求を通知する(④)．

提案方式 2：図 4に示すように，まずリソースが豊富な端末(A)は，リソースの余裕度(例えば余裕率や余裕量)を表す値が端末内に予め設定された閾値を超えた場合に，他端末(端末(B)～(D))からサービス MA が移動可能な状態であると判断し，コーディネータを介して他端末に広告する(①)．広告を受けた端末は，広告に含まれる端末(A)の余裕度と自端末の余裕度とを比較し，交渉をすべきか判断する(②)．交渉を行うと判断した場合には，その端末と直接交渉を行い，移動するか否かを決める(③)．

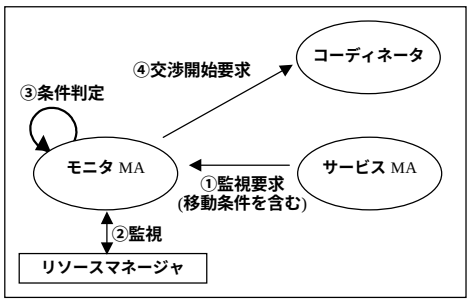

図 3 自端末の変化による交渉開始

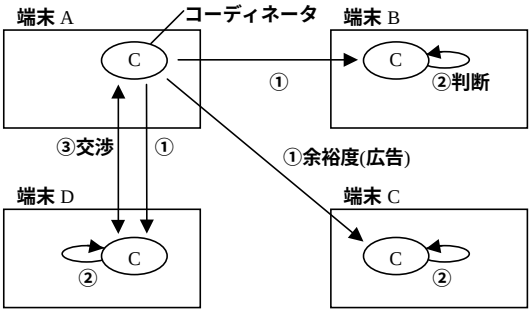

図 4 他端末の変化による交渉開始

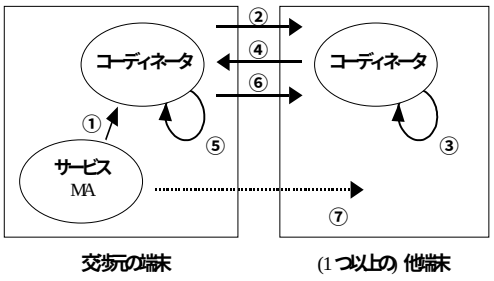

図 5 移動先決定のための交渉の流れ

提案方式 1 だけでは，他端末のリソースを有効に活用することはできない．例えば，他にリソースが豊富な端末が存在しても，それを他端末に通知する仕組みがなければ，他端末に存在するサービス MA は，リソース不足等により移動条件が満たされるまで，リソースが豊富な端末を発見できない．提案方式 2 を併せて用いることにより，この問題が解決できる．

4.3. 移動先決定のための交渉処理

提案方式では，交渉処理に用いるプロトコルとして，複数のエージェント間のタスク割り当てに良く利用されている"契約ネットプロトコル"[5]を用いている．契約ネットプロトコルは"タスク告示"，"入札"，"落札"の 3 段階の交渉プロトコルから構成される．

図 5の例を用いて，サービス MA の移動先を決定する流れを説明する．まず，自端末の状況に変化がおき，予め定義された移動条件が満たされると，サービス MA はコーディネータに他端末との交渉開始を要求する(①)．この要求を受けて，コーディネータは移動先決定のための条件を他端末にマルチキャストする(②)．ここで，マルチキャストされる範囲は予め定められた"交渉エリア"に限定される．各端末のコーディネータは自端末の状況を調査し(③)，その結果を交渉要求元に返却する(④)．交渉要求元のコーディネータは予め定められた"評価基準"(例えば，予測される処理性能や通信コストなど)に基づいて最も適切な移動先端末を選択し(⑤)，その結果を選択した端末を含む全ての端末に報告する(⑥)．これにより，適切な端末にサービス MA が移動する(⑦)．このような交渉プロトコルを用いることで，最適な端末を効率的に選定できる．

サービス MA をリソースが豊富な他端末に移動させても，サービス MA を利用するアプリケーション自体の処理効率が必ずしも向上するとは限らない．例えば，サービス MA とアプリケーション間の通信処理やサービス MA の移動処理などのオーバヘッドが発生するような場合には，他端末のリソースを利用しても処理効率の向上が望めないことがある．そのため，これらのオーバヘッドを考慮して評価基準を定義する必要がある．アプリケーションの処理性能の向上を目的とする場合，例えば以下のような評価式 fn (n は端末を表す添え字)を交渉に用いる．

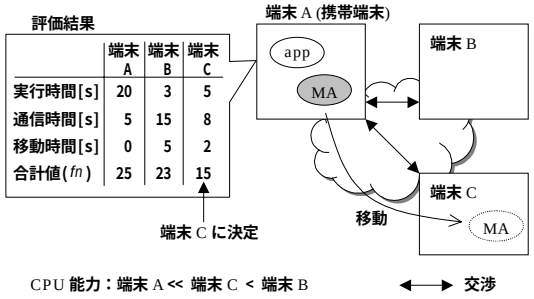

図 6 提案方式 1 による移動制御

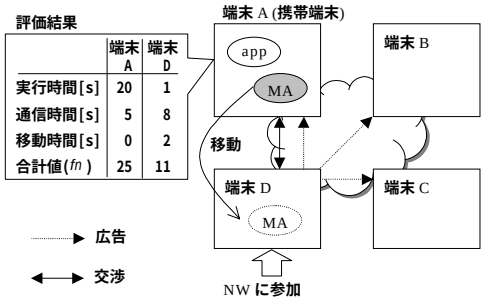

図 7 提案方式 2 による移動制御

fn = MA の実行時間の推定値
＋MA のデータ通信時間の推定値
＋MA の移動時間の推定値 (1)

ここで，右辺の第 1 項は計算リソース量(CPU 能力，メモリなど)，第 2 項はデータ通信量と通信帯域幅，第 3 項はサービス MA 自身のサイズと通信帯域幅等のパラメータから推定する値である．提案方式では，交渉時に推定を行うためのパラメータを各端末から収集し，評価式(1)を用いて各端末の処理時間を推定する．交渉の結果，例えば図 6に示すような評価結果が得られた場合，サービス MA は fn が最小となる端末 C に移動する．この例では，端末 B は端末 C よりも CPU 能力が高く，サービス MA の処理時間を最も短縮するが，アプリケーションとの通信処理や移動処理にかかる時間が長いため，評価式の値は端末 C に比べ大きくなる．これは，CPU 能力だけではアプリケーションの処理性能を向上させる端末を正確に選定できないことを示している．

図 7は，CPU 能力に余裕のある端末 D が新たにネットワークに参加した状況を示す．このとき，端末 D は他端末 A～C に対して，CPU 能力等の余裕値を広告する．サービス MA を実行している端末 A は，この広告を受け，図上示すように自端末と移動先端末における評価値を比較した結果，移動させた方がサービス MA の処理効率をより高めることができると判断した場合は，端末 A と端末 D との間で交渉を行い，端末 A から端末 D へサービス MA を移動させる．

5. 計算機シミュレーション

5.1. シミュレーションモデル

シミュレーションモデルは，ネットワーク上に PDA が 1 台，ノート PC が 1 台，PC サーバが 2 台接続されており，各パラメータの値が表 1 で与えられる環境を想定する．各端末の使用リソース(負荷)は時間的に変動する．したがって，MA が使用できるリソースもそれに応じて変動する．負荷変動モデルとして，各端末の総リソースを 1/10 に量子化したリソース単位で負荷を変動させる，ただし，時間毎の負荷の増減は量子化単位及び増減なしの 3 状態(±量子化単位，±0)に制限し，状態変化の確率をそれぞれ 2/5(増加), 2/5(減少), 1/5(変化なし)とする．例えば，PC サーバの場合，リソースの量子化単位は 2 であり，負荷変動は，10→8(減少)→8(変化なし)→10(増加) →12(増加)のように変化する．この環境において，PDA 上で実行されているアプリケーションに着目し，その処理時間を短縮するように MA の移動を制御する．また，MA の実行時間は，タスクサイズ÷リソース割当量で導出可能なモデルを仮定する．例えば，タスクサイズが 30000 の MA を半分実行した時点で端末のリソース割当量が 10 から 12 に変動したとすると，MA の実行時間は 15000÷10＋15000÷12＝2750[ms]となる．

ところで，表 1 中の移動条件・通知条件は交渉開始のトリガを決める閾値であり，MA の滞在する端末のリソース割当量が移動条件(ここでは 2 とする)よりも小さくなった場合，移動のための交渉が開始される．また，端末のリソース割当量が通知条件(ここでは 18 とする．この例ではノート PC(リソース割当量は最大 10)からの通知は行わない)以上の場合，他端末に移動の交渉を開始するように伝える．

シミュレーションは，MA がアプリケーションからのリクエストを受けて処理を実行し，レスポンスを返すまでを 1 回の処理として，これを 100 回繰り返して行う．なお，各処理の途中での移動はできないものとする．

表 1 計算機シミュレーション諸元

パラメータ	値
エージェントのタスクサイズ [ms]	30000
PC サーバのリソース割当量の最大値	20
ノート PC のリソース割当量の最大値	10
PDA のリソース割当量の最大値	1
メッセージ転送時間 [ms]	10
エージェントの移動時間[ms]	1000
移動先決定のための交渉時間 [ms]	1000
エージェントの移動条件	2
他端末への通知条件	18
エージェントの実行回数 [回]	100
端末のリソース割当量の変動間隔[ms]	1000

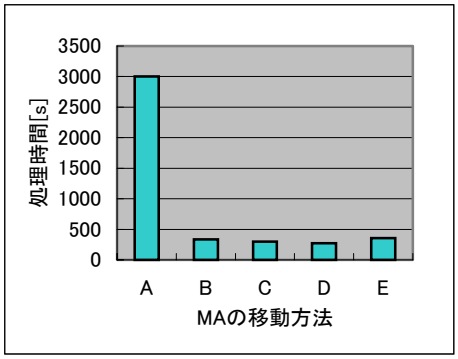

図 8 各移動方法における処理完了時間

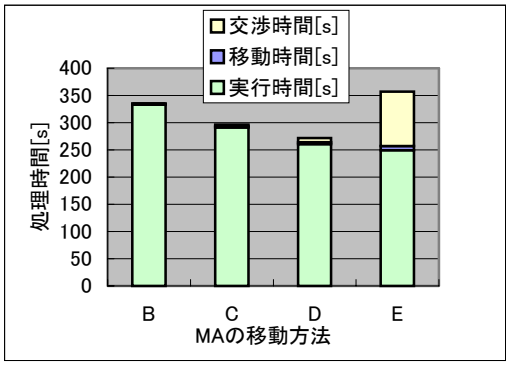

図 9 各移動方法における処理時間の内訳

5.2. 提案システムの適用効果

提案システムの適用効果を検証するため，5.1 で述べたモデルに基づき処理時間についてのシミュレーションを行った．その結果を図 8に示す．比較のために，提案方式を用いない場合の結果についても併せて示す．ここで，A～E は MA の移動方法(移動しない場合を含む)を示しており，(A) PDA で実行し続けた場合(移動しない場合), (B)最初の 1 回だけ交渉・移動しそれ以降は移動しない場合，(C)移動条件のみをトリガとして交渉を開始する場合(提案方式 1)，(D)移動条件＋通知条件をトリガとして交渉を開始する場合(提案方式 1＋2)，(E)定期的に交渉を行う場合，である．図より，MA が移動する場合(B～E)の処理が早く完了していることが分かる．

次に，B～E の各方法について処理時間の内訳を図 9に示す. 図より, 定期的に交渉する E は，MA の実行時間を大幅に減らすことはできるが，交渉の回数が多くなるため，処理時間が長くなる．これに対して，提案方式を用いた場合(C, D)には，交渉回数の増加を低く抑えつつ，実行時間の短縮を実現できる．特に D の場合には，提案方式 2 を用いることにより，他端末のリソースを有効に活用できることから，処理時間をより低減できることが分かる．

5.3. 移動処理･交渉処理による影響

次に，提案する移動制御方式においてオーバヘッドとなる MA の移動時間と端末間の交渉時間が処理時間に与える影響について調べるため，この 2 つの値をパラメータとして変化させた場合の処理時間をシミュレーションにより求めた．その結果を図 10，図 11に示す．図 10，図 11 はともに，移動時間を *x* 軸，交渉時間を *y* 軸，処理時間を *z* 軸として表現している．図より，移動方法 C，D ともに移動時間・交渉時間が短いほど処理時間が短くなっていることが分かる．以下，移動時間・交渉時間それぞれの影響について考察する．

まず，移動時間による影響について考察する．提案システムでは，交渉処理の中で MA の実行時間と移動時間を考慮した評価式を用いて，MA の移動を制御している．この動作を確認するため，移動時間の値を変えてシミュレーションを行った(交渉時間は 100ms で固定)．結果を図 12，図 13に示す．図より，移動方法 C，D ともに移動時間が長い場合は移動回数が少なく，移動時間が短い場合は回数が多くなっていることが分かる．これは，移動時間が短い場合は移動によって処理時間が短縮される可能性が高くなるが，移動時間が長くなると処理時間が短縮される可能性が低くなるためである. これより，提案方式が移動時間を考慮して適切に MA の移動を制御していることが分かる．さらに，移動方法 D では移動時間が長い場合は C と同程度であるが，移動時間が短い場合には C より多く移動することによりリソースを効率的に利用し，処理時間の一層の短縮が実現できる．

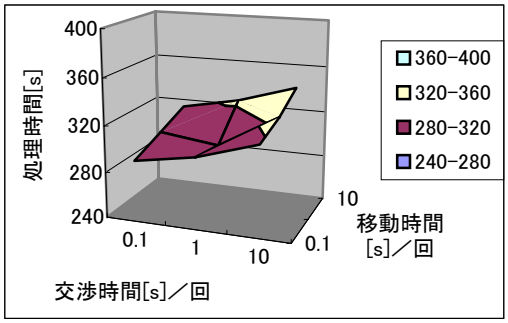

図 10 オーバヘッドによる影響(移動方法 C)

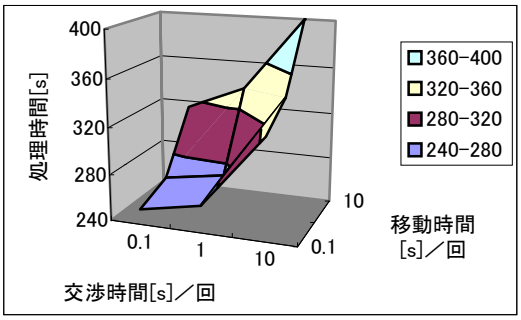

図 11 オーバヘッドによる影響(移動方法 D)

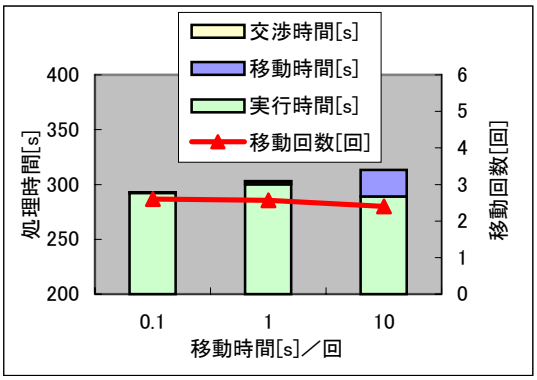

図 12 移動時間による影響(移動方法 C)

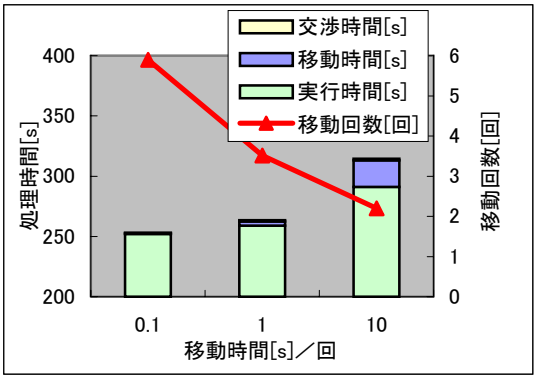

図 13 移動時間による影響(移動方法 D)

次に，交渉時間による影響について考察する．今回の提案方式では，交渉の回数はリソース割当量の変動によって決まり，交渉時間を考慮した交渉回数の制御は行っていない．このため，交渉時間が長い場合には，処理時間に占める交渉時間の割合が大きくなり，MA の移動による処理時間改善の効果が打ち消される．つまり，交渉時間はシステムの適用限界を決める値となる．交渉時間は交渉プロトコルの実装に依存するので，交渉時間が短くなるように交渉プロトコルを設計する必要がある．また，交渉時間に合わせて交渉頻度を変える等の工夫を加えることによって，交渉によるオーバヘッドを抑えることができると考えられる．

6. まとめ

本稿では，筆者らが提案したモバイルエージェントを基盤とするミドルウェアで導入したエージェント移動制御方式について述べ，その適用効果および移動制御に伴うオーバヘッドの影響を，シミュレーション結果をもとに示した．シミュレーション結果から，自端末と他端末の環境変化をトリガとして交渉を開始する提案方式の効果，および移動処理・交渉処理のオーバヘッドの影響を明らかにした．

謝　辞

日頃ご指導頂く当社情報通信研究所ワイヤレスシステム部 藤井輝也 部長に感謝いたします．

文　献

[1] K. Hiroshige, K. Kawakami, H. Sasaki, Y. Okataku, and S. Honiden, "Mobile Agent-Based Middleware for Mobile Terminals," http://computer.org/dsonline/0107/features/hir0107.htm

[2] 川上 憲治，広重 一仁，佐々木 宏，岡宅 泰邦，本位田 真一，"モバイル環境向けエージェント移動制御，" 情報処理学会 MBL 研究報告，2002 年 3 月．

[3] 川上 憲治，広重 一仁，吉岡 信和，本位田 真一，"モバイル環境向けエージェント移動制御 (その 2)，" 情報処理学会 MBL 研究報告，2002 年 5 月．

[4] 広重 一仁，川上 憲治，本位田 真一，"モバイルミドルウェアにおけるエージェント移動制御，" マルチメディア，分散，協調とモバイル(DICOMO2002)シンポジウム，2002 年 7 月．

[5] R. G. Smith, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver," IEEE Transactions on computers, vol.c-29, no.12, 1980.