

## 変数評価機能を有するモバイルエージェントを用いた 格子状に接続されたユビキタスコンピュータ群の制御

國本 慎太郎<sup>†1</sup> 藤田 直生<sup>†1</sup> 佐野 渉 二<sup>†2</sup>  
寺田 努<sup>†1,†3</sup> 塚本 昌彦<sup>†1</sup>

筆者らの研究グループでは、格子状に配置したユビキタスコンピュータ群に対して、それぞれのユビキタスコンピュータに組み込まれた入出力デバイスをモバイルエージェントを用いて制御する手法を提案している。モバイルエージェントを実行すべきコマンドの羅列であると捉え、モバイルエージェントの移動、並列処理、入出力制御などの機能をもつコマンドを作成した。本稿では、変数に関するコマンドを改良し、新たに変数評価コマンドを導入した。変数コマンドの改良により各コンピュータに固有の変数を設定でき、変数評価コマンドにより各コンピュータでの条件分岐や再帰処理ができるようになった。

### Controlling Ubiquitous Computers in Grid Topology using Mobile Agents including Variable Evaluation Function

SHINTARO KUNIMOTO,<sup>†1</sup> NAOTAKA FUJITA,<sup>†1</sup>  
SHOJI SANO,<sup>†1</sup> TSUTOMU TERADA<sup>†1</sup>  
and MASAHICO TSUKAMOTO<sup>†1</sup>

We have proposed a method using mobile agents to control I/O devices on ubiquitous computers in grid topology. By using a mobile agent program consisting of a set of commands, a user can perform various functions, such as migration, parallel processing, and I/O control by using these commands. In this paper, we improved the commands about variables, and incorporate variable evaluation commands. Therefore, user can utilize local variables, conditional branch and recursive processing.

#### 1. はじめに

近年、情報機器の小型、軽量化、低価格化に伴い、小型のコンピュータが生活環境のいたるところに埋め込まれたユビキタスコンピューティングの実現が期待されている。ユビキタスコンピューティング環境では多数のコンピュータを制御することが望まれるため、個々のコンピュータへのプログラミングよりコンピュータ群全体へのプログラミングが求められる。筆者らの研究グループでは、格子状に配置したユビキタスコンピュータ群に対して、それぞれのユビキタスコンピュータに組み込まれた入出力デバイスをモバイルエージェントを用いて制御することで、ユビキタスコンピュータ群全体の入出力制御を行う手法を提案している<sup>1)</sup>。モバイルエージェントを実行すべきコマンドの羅列であると捉え、モバイルエージェントの移動、並列処理、入出力制御などの機能をもつコマンドによって、並列処理や環境内のコンピュータ数の変化へ対応できるようにした。本稿では、変数に関するコマンドを改良し、新たに変数評価に関するコマンドを導入した。このことにより各コンピュータ固有の変数と全コンピュータに対する共通の変数を設定できるようになり、それらの値を評価することにより、各コンピュータでの条件分岐や全コンピュータを対象とした再帰処理ができるようになった。

以下、2章で関連研究について説明し、3章でユビキタスコンピュータ群を制御するためのモバイルエージェントについて述べ、4章でモバイルエージェントの使用例について説明し、最後に5章で本研究をまとめる。

#### 2. 関連研究

寺田らはルールに基づく入出力制御という枠組みのユビキタスシステムを提案するとともに、その動作記述言語および入出力制御を行うユビキタスコンピュータの設計、実装を行った<sup>2)</sup>。センサまたは単一の制御アーキテクチャに基づいて、アクチュエータを使用してルールの追加、変更をすることにより、さまざまなユビキタスコンピューティング環境を柔軟に構築できる。しかし、ルールを用いることにより記述に制限がある、また複数のデバイスで

<sup>†1</sup> 神戸大学大学院工学研究科  
Graduate School of Engineering, Kobe University

<sup>†2</sup> 神戸大学大学院自然科学研究科  
Graduate School of Science and Technology, Kobe University

<sup>†3</sup> 科学技術振興機構さきがけ  
PRESTO, Japan Science and Technology Agency

協調して制御をおこなう際、各ユビキタスコンピュータにルールを記述する必要があるなどの問題がある。ユビキタスコンピュータ群の制御の研究として、多数のコンピュータをあたかも1つのユビキタスコンピュータを扱うような記述で制御するマクロプログラミングがあり、多くの研究がされている<sup>3)-5)</sup>。たとえば、Kairos<sup>3)</sup>は、コンピュータ群に対して、複数のコンピュータに及ぶ処理やコンピュータ間のトポロジを用いた処理を1つのプログラムで記述できる。しかしこれらのシステムはコンピュータ群に対するプログラムを個々のコンピュータが処理するバイナリに変換して分配する機構を有しているが、プログラムを変更する度に各コンピュータにバイナリを分配しなければならず、状況に応じて処理内容をシームレスに変更することは困難であり、デバイスの数に対するスケーラビリティがない。モバイルエージェントに関する研究も多数行われている<sup>6)-9)</sup>。Agilla<sup>6)</sup>ではプログラマはミドルウェアによって提供されるAPIを利用してセンシングデータを取得するプログラムを記述する。これにより、プログラムを動的にノードに配備することができ、センシング情報に応じてプログラムを動的に配備可能である。Yu-Chee Tseng<sup>7)</sup>らや Yingyue Xu<sup>8)</sup>らは無線センサネットワークにおいてモバイルエージェントを用いてターゲットを追跡する手法を提案している。格子状ネットワーク(2D mesh)は、特に無線センサネットワークおよびメディア・アクセス制御の分野で、主に研究されている<sup>10),11)</sup>。一般的に、格子状ネットワークのような規則的な構造は均一なノードのパターンを提供し、ローカライズされたプログラムはパターンを繰り返すことによってグローバルに展開することができる。更に、格子状ネットワークは、与点からのホップを数えることによりノードの位置を計算することを簡単にする。WAVEは、分散コンピュータ上のモバイル・エージェント機能性を提供している<sup>12)</sup>。ネットワーク用の一般的な接続を仮定し、移動する際はリンクの名前に依存する。したがって、大規模ネットワークの中では移動する際、リンク名の管理が特に制限のある資源を備えたノード上では困難である。

### 3. モバイルエージェントを用いた制御手法

#### 3.1 アプローチ

本研究では、隣接するユビキタスコンピュータとの通信機能をもつ、環境内に数百から数千個で存在するユビキタスコンピュータ群を制御することを想定する。各ユビキタスコンピュータが全体の中でどのようにつながっており、全体でどのような位置にあるかを知り、1つのプログラムをローカルなトポロジを利用して各方向に移動させることにより環境内のユビキタスコンピュータ全体を対象としたプログラミングを行えるような方式を確立する。

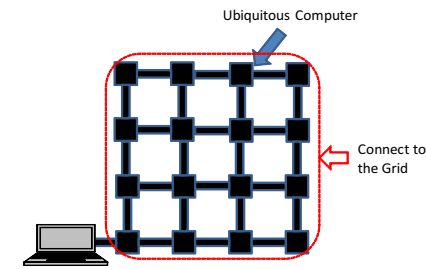


図1 想定環境

そのため、図に1に示すように、グローバルとローカルが明白な格子状ネットワーク内のコンピュータにモバイルエージェントのフレームワークを適用する。格子状ネットワークを用いることで、モバイルエージェントが移動する際、その位置や移動経路を理解するのが容易になる。モバイルエージェントを実行すべきコマンドの羅列であると捉え、ユビキタスコンピュータ群に対する入出力制御を行うためさまざまな種類のコマンドを作成した。コマンドのほとんどをアルファベット数字や記号で表すことでメモリ消費量を抑えた。

しかし、これまでは変数情報に応じて入出力制御を変更することができなかった。そこで本稿では変数コマンドを改良し、新たに変数評価コマンドを導入した。これにより各コンピュータ固有の変数と全コンピュータに対する共通の変数を設定でき、それらの値を評価することで、各コンピュータでの条件分岐や全コンピュータを対象とした再帰処理ができるようになった。

#### 3.2 コマンド

作成したコマンドの種類、動作について説明する。ここで  $n$  は0から255の整数、 $m$  は0から65525の整数、 $v$  は変数で  $g$  から  $z$  までの20種類のアルファベットを用い、 $w$  は配列で  $a$  から  $f$  までの6種類を用い、 $Z_m$  は任意のコマンド列とする。

##### 3.2.1 進行, 進路変更, 方向分岐コマンド

表1に進行, 進路変更, 方向分岐コマンドを示す。このコマンドは、格子状ネットワーク内のユビキタスコンピュータにおいてモバイルエージェントの前後, 左右の移動の制御するためのコマンドである。進行コマンド「F」は「F」より後ろのコマンド列を現在の進行方向先のユビキタスコンピュータに移動する(以降この動作を「進む」と表記する)コマンドである。進路変更コマンドの「R」, 「L」, 「B」は現在の進行方向をそれぞれ右, 左, 後ろに変更するコマンドである。方向分岐コマンドの「(」は、進行方向先にユビキタスコン

表 1 進行, 進路変更, 方向分岐コマンド

F	進行方向に「F」より後ろのコマンド列を送る.
R	進行方向を右に変更する.
L	進行方向を左に変更する.
B	進行方向を後ろに変更する.
$(Z_1; Z_2; \dots; Z_m) Z_{m+1}$	コマンド列を進める方向へ送る

表 2 入出力制御コマンド

$Ck$	フルカラー LED を $k$ が指定する色に発光させる.
$C(n_1; n_2; n_3)$	$n_1, n_2, n_3$ の部分にそれぞれ r 値, g 値, b 値が入り, 値を指定して発光する.
$Mk_1 v$	赤外光センサまたは可視光センサの値を $v$ に代入する.

表 3 繰り返しコマンド

$m(Z_1)$	$Z_1$ を $m$ 回繰り返す.
$*(Z_1)$	$Z_1$ を無限に繰り返す.

コンピュータがつながっているかを判断し, 自律的に進行, 進路変更を行うコマンドである. 現在の進行方向から  $Z_1$  の進行, 進路変更コマンドの示す方向にユビキタスコンピュータがつながっているかを調べ, その方向に進行可能なら  $Z_1 Z_{m+1}$ , 進行不可能なら  $Z_2 Z_{m+1}$ ,  $Z_1$ ,  $Z_2$  方向に進行不可能なら  $Z_3 Z_{m+1}$ ,  $\dots$  と順番に評価して実行できれば実行する. どれか 1 つを実行するもしくは  $Z_m$  まで不可能ならばそこで処理を終了する.

### 3.2.2 入出力制御コマンド

表 2 に入出力制御コマンドを示す. 出力制御コマンド「C」は LED を指定の色に発光させる. 「 $k$ 」の部分には r, g, b, m, y, c, w が入り, それぞれ赤 (red), 緑 (green), 青 (blue), 紫 (magenta), 黄 (yellow), シアン (cyan), 白 (white) に発光する. 「 $C(n_1; n_2; n_3)$ 」は  $n_1, n_2, n_3$  の部分にそれぞれ r 値, g 値, b 値が入り, 値を指定して発光する. 入力制御コマンド「M」は搭載されている可視光センサ, 赤外光センサの値を変数に代入する.  $k_1$  は「i」もしくは「v」であり「i」ならば赤外光 (invisible) センサ, 「v」ならば可視光 (visible) センサの値を変数  $v$  に代入する.

### 3.2.3 繰り返しコマンド

表 3 に繰り返しコマンドを示す. このコマンドはコマンド列の繰り返し処理を容易にするためのコマンドである.  $Z_1$  を  $m(Z_1)$  とすると  $Z_1$  を  $m$  回繰り返す,  $*(Z_1)$  とすると  $Z_1$  を

表 4 特殊コマンド

$Wm$	次のコマンドの処理を行うまで $m \times 100\text{ms}$ 待機する.
$Pm$	LED の発光時間を調整する. LED が $m \times 100\text{ms}$ 間発光し, 消える. $m \times 100\text{ms}$ 間待機してから次のコマンドの処理を行う.
:	コマンドを区切る. 処理は何も行わない.
K	プログラムの強制終了を行う.

表 5 変数制御コマンド

$Svn$	$v$ を $n$ に設定する.
$Sv(Z_m)$	$v$ を $Z_m$ に設定する
$Sum_1; n_2$	$w$ の $n_1$ 番目を $n_2$ に設定する.
$Ivn$	$v$ を $n$ 増加させる.
$Dvn$	$v$ を $n$ 減少させる.
$Evk_2 n(Z_1; Z_2) Z_3$	$v$ が $k_2 n$ で指定した条件を満たせば $Z_1 Z_3$ を, そうでなければ $Z_2 Z_3$ を実行する.
$Uv$	変数 $v$ に書き換える.

無限に繰り返す.

### 3.2.4 並列処理コマンド

このコマンドはエージェントを分岐させることにより複数のコンピュータにおける並列処理を可能にするためのコマンドである. タスクを  $Z_1, Z_2, \dots, Z_m$ , 全タスクに共通する処理を  $Z_{m+1}$  とすると, 「 $[Z_1; Z_2; \dots; Z_m] Z_{m+1}$ 」により  $Z_1 Z_{m+1}, Z_2 Z_{m+1}, \dots, Z_m Z_{m+1}$  を同時に実行できる.

### 3.2.5 特殊コマンド

表 4 に特殊コマンドを示す. 「 $Wm$ 」は処理のタイミングを調節できるコマンドである. 「 $Pm$ 」は LED の発光時間を調節できるコマンドである. LED の発光コマンドと組み合わせると LED が  $m \times 100\text{ms}$  間発光し, 消える. そしてその状態のまま  $m \times 100\text{ms}$  間待機して次のコマンドの処理を行う. 「:」はコマンドや数字を区切るコマンドである. 数字が二つ並んだときにそれを 1 つの数字で処理を行わないようにする. 「K」はプログラムの強制終了を行うコマンドである.  $KZ_1$  とすると K 以降の  $Z_1$  は実行されない. このコマンドにより無限繰り返しの強制終了が可能である.

### 3.3 変数コマンド

表 5 に改良, 新たに導入した変数に関するコマンドを示す. 変数は 2 種類あり, 変数名 a から c と, g から w はエージェントが保持し, 他のコンピュータに移動しても値を利用でき

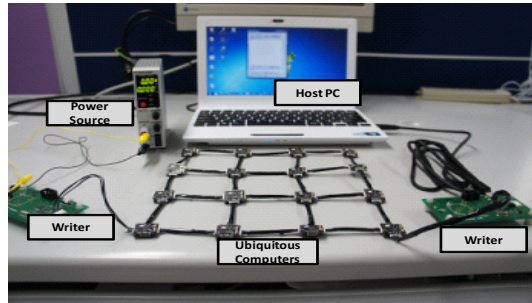


図 2 システム構成

る変数（以下グローバル変数）であり、変数名  $d$  から  $f$  と、 $x$  から  $z$  はエージェントが保持せず、他のコンピュータに移動しない変数（以下ローカル変数）である。変数設定コマンド「S」は「 $Sv_n$ 」で  $v$  を  $n$  に設定する。また「 $Sv_{v_1}$ 」で変数  $v_1$  の値を  $v$  に設定することができ、「 $Sv(Z_1)$ 」で変数  $v$  に  $Z_1$  を設定する。「 $Sw_{n_1:n_2}$ 」で配列  $w$  の  $n_1$  (以降  $w(n_1)$  と表記する) を  $n_2$  に設定する。配列は 10 まで指定できるようにした。 $n_1$  としては 0 から 9 の整数を使用する。「 $Sw:n_2$ 」と表記すると、配列  $w$  の最後尾に  $n_2$  を代入する。た、「 $Swv:v_1$ 」と表記すると、配列  $w$  の  $v$  番目を変数  $v_1$  の値に設定でき、「 $Swv:w_1n_1$ 」と表記すると、 $w(n)$  を、 $w_1(n_1)$  値に設定できる。変数増減コマンド「I」、「D」は「 $Iv_n$ 」で変数  $v$  の値を  $n$  増加させ、「 $Dv_n$ 」変数  $v$  の値を  $n$  減少させる。変数評価コマンドとして「E」は変数の値を評価するものである。変数  $v$  の値と  $k_2n$  の部分で設定した条件を評価する。 $k_2$  には e, n, g, l, または b が入る。それぞれ  $n$  と読み取った値が等しい (equal),  $n$  と読み取った値が等しくない (not equal), 読み取った値が  $n$  より大きい (greater), 読み取った値が  $n$  より小さい (less) ことを示す。b のときは「 $Evb_{n_1:n_2}(Z_1;Z_2)Z_3$ 」と表記することにより読み取った値が  $n_1$  より大きく、 $n_2$  より小さい (between) ことを示す。条件を満たせば  $Z_1Z_3$ , そうでなければ  $Z_2Z_3$  を実行する。「U」は変数  $v$  に書き換えるコマンドである。変数  $v$  に  $n$  を設定していれば、「 $Uv$ 」は  $n$  に書き換えられる。変数  $v$  にコマンド列「 $Z_m$ 」を設定していれば「 $Uv$ 」は  $Z_m$  に書き換えられる。このコマンドは  $v$  には変数として  $v$  から  $z$  のいずれかを使用する。これら変数評価コマンドとグローバル、ローカル変数を組み合わせることにより各コンピュータでの条件分岐や、コンピュータでの再帰処理ができるようになった。

### 3.4 実装

前節で述べたのコマンドを扱えるモバイルエージェントを用いて、ユビキタスコンピュー

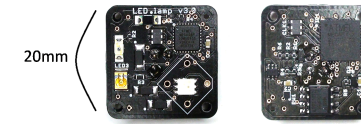


図 3 制御対象の小型デバイス

タ群を制御するための実行環境を構築した。図 2 にシステム構成を示す。制御対象であるユビキタスコンピュータ、ユビキタスコンピュータの接続線、ユビキタスコンピュータの書き込み機（ライター）、シリアル通信用アプリケーションは筆者らの研究グループで作成したものである。USB ケーブルとライターを接続し、ライターと格子状のユビキタスコンピュータ群を接続する。電源を同じく別のライターに接続してライターから格子状のユビキタスコンピュータ群と接続する。制御対象のユビキタスコンピュータとしては図 3 に示す筆者らの研究グループで開発した小型デバイスを用いる。大きさは 20mm 四方で、マイコンには Atmel 社製ワンチップマイコンである AVR を使用している。フルカラー LED、可視光センサ、赤外光センサが搭載され、通信線で隣接するユビキタスコンピュータと 4 方向まで通信できる。

### 3.5 モバイルエージェントの処理

ここではモバイルエージェントの処理の流れについて説明する。まず、ユビキタスコンピュータにはモバイルエージェントの処理を行うために、あらかじめコマンドを解釈して実行するエンジンを格納させる。各ユビキタスコンピュータはモバイルエージェントを格納するメモリ（以下コマンドメモリ）をもっており、そこに格納されたモバイルエージェントを順に実行していく。PC から送信すると、最初に PC とつながっているユビキタスコンピュータのコマンドメモリに格納される。このとき、図 4 のようにユビキタスコンピュータが下の通信ポートからモバイルエージェントを受信するとき、モバイルエージェントは上方向の向きを持つ。同様に PC からモバイルエージェントを受信する通信ポートが上、左、右のときはそれぞれ下、右、左の進行方向をもつ。

まず進路変更、入出力制御、変数設定、増減、終了コマンドを除く特殊コマンドの処理の流れについて説明する。ここで、 $Z_k$  は任意のコマンド列とする。これらのコマンドを  $Y_m$  とした、 $Y_mZ_1$  において、 $Y_m$  は受信したユビキタスコンピュータにて実行され、次の  $Z_1$  の最初のコマンドの処理を行う。

次に進行、終了、繰り返し、並列処理、方向分岐、変数評価コマンドについて説明する。

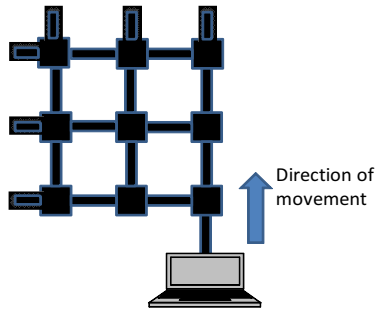


図 4 格子状のコピキタスコンピュータ群

エージェントの処理のうち、これらのコマンドは、コマンド文字列を直接書き換えることによって実現している。具体的には、表 6 のように書き換えることによって、これらの制御構文が実現されている。以下、書き換えについて説明する。

#### 進行コマンドの処理

進行コマンド  $FZ_1$  の処理について示す。F が認識されると F 以降のコマンド列  $Z_1$  は進行方向先のコマンドメモリに格納されそこで実行される。移動する際はグローバル変数の情報も一緒に移動する。もし、グローバル変数が存在すれば、コマンド列は  $Sv_1Sv_1n_1 \cdots Sv_{m_1}n_2Sw_1n_3:n_4 \cdots Sv(Z_k)Sv_1(Z_{k+1}) \cdots Z_1$  と書き換えられ進行方向先へ移動する。この処理により移動先でもグローバル変数を使うことができる。

#### 終了コマンドの処理

終了コマンド  $KZ_1$  の処理について示す。K が認識されるとそれ以降のコマンド列  $Z_1$  がコマンドメモリから消去される。これによりプログラムの終了ができる。

#### 繰り返しコマンドの処理

繰り返しコマンド  $r(Z_1)Z_2$  の処理について示す。「r」は整数または「\*」のいずれかである。「r」を認識するとコマンド列から  $Z_1$  を探す。「r」が整数  $n$  のときコマンドメモリは、 $Z_1n-1(Z_1)Z_2$  と書き換えられ、 $Z_1$  を 1 回行った後に  $n-1$  回繰り返し、 $Z_2$  を行う。「r」が「\*」のときも同様に  $*(Z_1)Z_2$  は、 $Z_1*(Z_1)Z_2$  と書き換えられる。

#### 並列処理コマンドの処理

並列処理コマンド  $[Z_1; Z_2; \cdots; Z_m] Z_{m+1}$  の処理について示す。[ を認識すると、コマンド列の中から  $Z_1$  と  $Z_{m+1}$  を探してコマンドメモリは  $Z_1Z_{m+1}$  に書き換えられ実行する。 $Z_1Z_{m+1}$

表 6 コマンドの処理

$\frac{Beforeconversion}{Afterconversion}$	処理の種類
$\frac{FZ_1}{Sv_1n_1 : n_2 \cdots Sv_{m_1}n_3 \cdots Z_1}$	グローバル変数とともに移動する。
$\frac{KZ_1}{}$	プログラムを強制終了する
$\frac{n(Z_1)Z_2}{Z_1n-1(Z_1)Z_2}$	有限繰り返し
$\frac{*(Z_1)Z_2}{Z_1*(Z_1)Z_2}$	無限繰り返し
$\frac{[Z_1; Z_2; \cdots; Z_m]Z_{m+1}}{Z_1Z_{m+1}; Z_2Z_{m+1} \cdots; Z_mZ_{m+1}}$	並列処理
$\frac{(Z_1; Z_2; \cdots; Z_m)Z_{m+1}}{Z_1Z_{m+1}orZ_2Z_{m+1}or \cdots orZ_mZ_{m+1}}$	方向分岐
$\frac{Evk_2n(Z_1; Z_2)Z_3}{Z_1Z_3orZ_2Z_3}$	変数評価
$\frac{UvZ_1}{nZ_korZ_kZ_1}$	変数評価
$\frac{Y_mZ_1}{Z_1}$	そのほかのコマンド

を実行していき、移動もしくは全処理が終了し、コマンドメモリが空になれば、同様に  $Z_2$  と  $Z_{m+1}$  を探して  $Z_2Z_{m+1}$  に書き換えられ実行する。以降同じ処理を  $Z_mZ_{m+1}$  を実行するまで繰り返す。

#### 方向分岐コマンドの処理

方向分岐コマンド  $(Z_1; Z_2; \cdots; Z_m) Z_{m+1}$  の処理について示す。( を認識すると、 $Z_1$  を探し、今の進行方向を記憶した状態で  $Z_1$  の先頭が示す方向の通信ポートにコピキタスコンピュータがつながっているかを調べる。つながっていれば  $Z_{m+1}$  を探してコマンドメモリは  $Z_1Z_{m+1}$  に書き換えられ実行し、 $(Z_1; Z_2; \cdots; Z_m) Z_{m+1}$  の処理を終了する。コピキタスコンピュータがつながっていれば  $Z_2$  の先頭が示す方向の通信ポートにコピキタスコンピュータがつながっているかを調べ同じ処理を行う。

#### 変数評価コマンド「E」の処理

変数評価コマンド  $Evk_2n(Z_1;Z_2)Z_3$  の処理の流れについて示す。「E」を認識すると、変数  $v$  の値と入力した値  $n$  を  $k_2$  の条件で評価を行う。条件を満たせば  $Z_1Z_3$  に書き換えられ、満たさなければ  $Z_2Z_3$  に書き換えられ実行する。

#### 変数評価コマンド「U」の処理

変数評価コマンド  $UvZ_1$  の処理の流れについて示す。「U」を認識すると、変数  $v$  の情報を評価する。変数  $v$  に  $n$  を設定していれば、 $UvZ_1$  は  $nZ_1$  に書き換える。変数  $v$  にコマンド列  $Z_k$  を設定していれば  $UvZ_1$  は  $Z_kZ_1$  に書き換えられる。このことにより、たとえば変数  $v$  に  $n$  を設定していれば、 $Uv(Z_1)$  は  $n(Z_1)$  に書き換えられ、 $Z_1$  を  $n$  回繰り返す。また、変数  $v$  に「 $Z_kUv$ 」を設定していれば  $Uv$  は  $Z_kUv$  に書き換えられ、 $Z_k$  を再帰的に繰り返すことができる。

### 4. モバイルエージェントの利用例

ここでは前述のコマンドを組み合わせたモバイルエージェントの使用例について述べる。トポロジを利用し、各ユビキタスコンピュータの入出力制御を行うことにより全体で1つの動作を行っていく。ここでは変数評価コマンドを用いた使用例を示し、このコマンドの有効性について考察する。

#### オブジェクトトラッキング

追跡対象者の真下が暗くなることを利用し、照度センサを用いて対象者を追跡していく。この処理は以下のプログラムにより実現できる。

[作成例] `*(MiyEyg200(W10Cr;[B;R;L;:]FMiyEyg200(;;K)))`

[説明] 図5に処理の流れを示す。もし対象者がユビキタスコンピュータの真下にいればそのあたりは暗くなるため、そのユビキタスコンピュータでは  $MiyEyg200$  が満たされ、 $W10Cr$  を1秒間隔で実行する。対象者が移動すれば  $MiyEyg200$  が満たされず、 $[B;R;L;:]F$  により4方向へ  $MiyEyg200(;;K)*(MiyEyg200(W10Cr;[B;R;L;:]FMiyEyg200(;;K)))$  を送る。受信先で対象者が存在しなければ  $MiyEyg200$  が満たされず、 $K$  が実行されるため処理を終了し、存在する場所では  $F$  が実行されるので同じ処理を続ける。

[実験] このモバイルエージェントを実際に実行し動作確認を行った。図6のように床に50 間隔にデバイスを配置し、対象者には図7の経路を歩いてもらった。プログラム中の  $Cr$  により、対象者が通った経路上のLEDは赤色に点灯する。図8に成功例と失敗例を

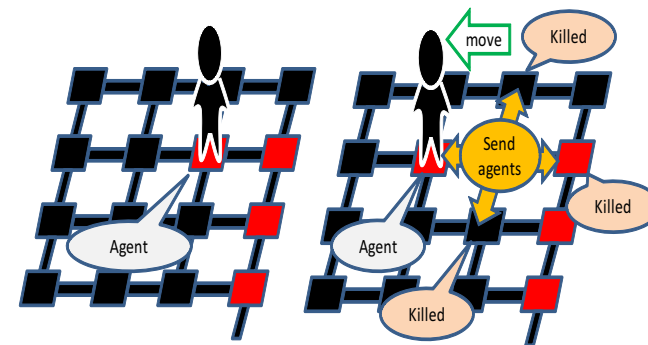


図5 処理の流れ



図6 作成した環境

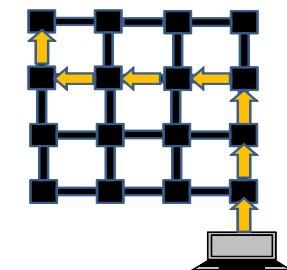


図7 移動経路

示す。正しく実行できたときは、図8(a)のように移動経路上のユビキタスコンピュータのLEDのみ赤色に点灯する。失敗した例では、図8(b)のように途中のユビキタスコンピュータで処理を終了している。この失敗が起こった原因としては、追跡対象者が1秒以内に2つ先のユビキタスコンピュータへ移動してしまったこと、または、対象者がユビキタスコンピュータ上を通過したにもかかわらずセンサ値が設定したしきい値(200)を超えなかったことが考えられる。これは、環境によってしきい値を変えること、 $W10$ の部分を  $W5$  などに変えること、また4方向だけでなくさらにその先にまであらかじめエージェントを送信しておくことにより解決できると考えている。

#### 日照時間測定

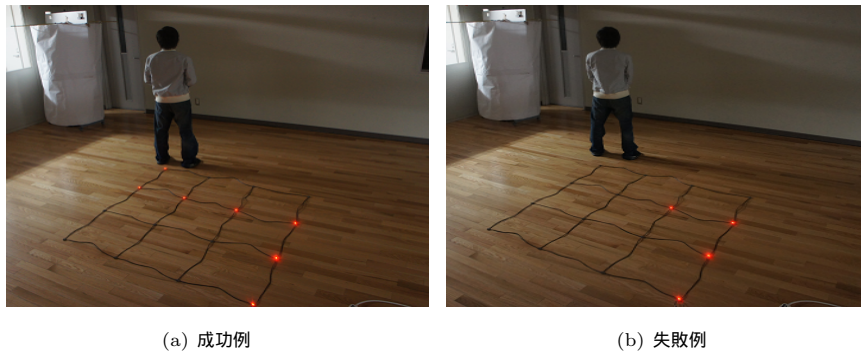


図 8 トラッキング結果例

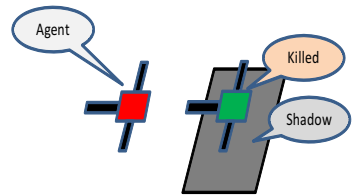


図 9 日照時間測定プログラムの処理

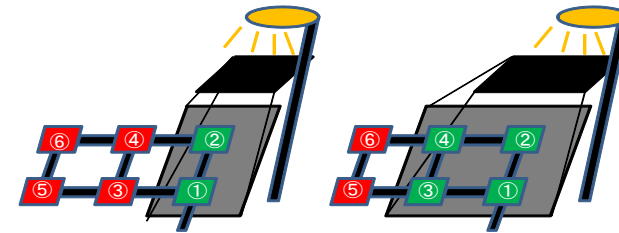


図 10 日照時間測定の実行環境

日陰の部分のが暗くなることを利用し、照度センサを用いてその部分が日向か日陰なのかを判断し、日照時間を測定する。この処理は以下のプログラムにより実行できる。

[作成例]  $*(MizEzgz200(CgK;CrIxEx60(Sx0Iy;:))W600)$

[説明] 図 9 に処理の流れを示す。もしコンピュータが日向にあれば、その部分は明るくなるため、そのコンピュータは MizEzgz200 は満たされないため、CrIxEx60(Sx0Iy;:))W600 を実行する。x を 1 増加させ、x が 60 になれば y を 1 に設定し、x を 0 に設定する。この処理を 1 分ごとに行うので x が分を表し、y が時を表す。コンピュータが日陰にあればその部分は暗くなるので CgK により、処理を終了する。最後に各コンピュータの x、y の値を確認する。なおこのプログラムで LED を点灯させるのは処理の継続、終了を視覚的に確認するためである。



(a) 45 分後の影の状態 (b) 65 分後の影の状態

図 11 時間による影の状態

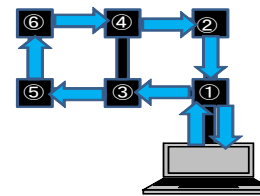


図 12 変数を調べるプログラムの移動経路

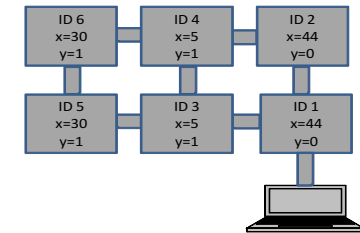
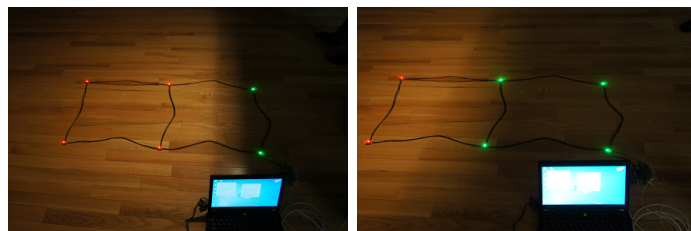


図 13 各コンピュータの変数の結果

[実験] このモバイルエージェントを図 10 のような仮想環境で実際に実行し、動作確認を行った。このモバイルエージェントプログラムを各コンピュータで実行するためプログラムの先頭に [FL;L][FF;F;:] を加える。このことにより各コンピュータへこのモバイルエージェントプログラムを送ることができる。コンピュータを 500mm 間隔にセットし、90 分間測定し、45 分後にコンピュータ 1 と 2 に、65 分後にコンピュータ 3 と 4 を図 11 のように影で覆った。90 分後、各コンピュータの変数を測定した。コンピュータの変数を確認するモバイルエージェントプログラムは  $L*(Sa;xSb;y(F;RF))$  により確認できる。このモバイルエージェントは図 12 のような経路で移動し、移動した先のコンピュータの変数を順番に配列 a,b に格納する。つまり、コンピュータ 1 の x,y は a(0),b(0) に、コンピュータ 6 の x,y はそれぞれ a(3),b(3) に格納される。図 13,14 に実行結果と各コンピュータの変数の値を示す。この結果により正しく動作ができたことがわかった。今後は実環境で長時間の測定を行い、動作確認を行う必要がある。

上述のようにユビキタス環境で想定されるユビキタスコンピュータ群の入出力制御をモバ



(a) 45 分後の結果

(b) 65 分後の結果

図 14 日照測定プログラムの実行結果

イルエージェントを用いて実現することができた。変数評価コマンドを導入することにより各コンピュータでの条件分岐が可能になり、さまざまな条件で各ユビキタスコンピュータに対するプログラミングができるようになり、それらが互いに連携し、ユビキタスコンピュータ群に対するユーザーの要求を達成することができると確認できた。

## 5. ま と め

本研究では、格子状に配置したユビキタスコンピュータ群に対して、それぞれのユビキタスコンピュータに組み込まれた入出力デバイスを制御するためのプログラミングを想定し、モバイルエージェントによる制御を行った。ユビキタスコンピュータにモバイルエージェントを用いるため、進行方向に移動、並列処理、入出力制御機能などをもつコマンドを多数作成した。本稿では新たに変数評価コマンドを導入することにより各コンピュータでの条件分岐が可能になり、さまざまな条件で各ユビキタスコンピュータに対するプログラミングができるようになり、それらが互いに連携し、ユビキタスコンピュータ群に対するユーザーの要求を達成することができると確認できた。

## 謝 辞

本研究の一部は、文部科学省科学研究費補助金基盤研究 (A)(20240009, 23240010) によるものである。ここに記して謝意を表す。

## 参 考 文 献

- 1) 國本慎太郎, 藤田直生, 佐野渉二, 寺田 努, 塚本昌彦: 格子状に接続されたユビキタスコンピュータ群のモバイルエージェントを用いた制御手法, マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO2011), pp. 741-748 2011.
- 2) T. Terada, M. Tsukamoto, K. Hayakawa, T. Yoshihisa, Y. Kishino, A. Kashitani, and S. Nishio: Ubiquitous Chip: a Rule-based I/O Control Device for Ubiquitous Computing, *Proceeding of the International Conference on Pervasive Computing (Pervasive 2004)*, pp. 238-253, 2004.
- 3) R. Gummadi, O. Gnawali, and R. Govindan: Macro-Programming Wireless Sensor Networks Using Kairos, *Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS2005)*, pp. 126-140, 2005.
- 4) R. Newton, G. Morrisett, and M. Welsh: The Regiment Macroprogramming System, *Proceedings of 6th International Conference on Information Processing in Sensor Networks (IPSN2007)*, pp. 489-498, 2007.
- 5) B. Urs and K. Gerd: RuleCaster: A Macroprogramming System for Sensor Networks, *Proceedings of the 21th Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA2006)*, 2006.
- 6) C.-L. Fok, G.-C. Roman and C. Lu: Mobile agent middleware for sensor networks: an application case study, *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN2005)*, pp. 382-387 2005.
- 7) Y.-C. Tseng, S.g-P. Kuo, H.-W. Lee and C.-F. Huang: Location Tracking in a Wireless Sensor Network by Mobile Agents and Its Data Fusion Strategies, *Proceedings of the Computer Journal*, Vol. 47, Issue 4, pp. 448-460, 2004.
- 8) Y. Xua, and H. Qi: Mobile agent migration modeling and design for target tracking in wireless sensor networks, *Ad Hoc Networks*, Vol. 6, Issue 1, pp. 1-16, 2008.
- 9) S. Ilarria, E. Mena, and A. Illarramendib: Using cooperative mobile agents to monitor distributed and dynamic environments, *Information Sciences*, Vol. 178, Issue 9, pp. 2105-2127, 2008.
- 10) C. Zhang and T. Herman: Localization in Wireless Sensor Grids, *Computers and Their Applications*, pp. 388-393, 2006.
- 11) J. M. Gutierrez Lopez, R. C. Rumin, T. M. Riaz, J. M. Pedersen and O. B. Madson: Characterization of Static/Dynamic Topological Routing for Grid Networks, *Proceedings of the 8th IEEE International Conference on Networks (ICN2009)*, pp. 367-373, 2009.
- 12) P.-S. Sapaty: Mobile processing in open systems, *Proceedings of the 5th IEEE High Performance Distributed Computing (HPDC1996)*, pp. 182-191, 1996.