M-82 ユビキタスコンピューティングのための入出力制御デバイスの動作記述言語

A Behavior Description Language of I/O Control Devices for Ubiquitous Computing

寺田 努 *1 Tsutomu Terada
塚本 昌彦 *2 Masahiko Tsukamoto
坂根 裕 *3 Yutaka Sakane
義久 智樹 *2 Tomoki Yoshihisa
岸野 泰恵 *2 Yasue Kishino
早川 敬介 *4 Keisuke Hayakawa
柏谷 篤 *4 Atsushi Kashitani
西尾 章治郎 *2 Shojiro Nishio

1. はじめに

いつでもどこでもコンピュータにアクセスできる世界であるユビキタスコンピューティング環境を実現するため，筆者らの研究グループでは，ルール形式で動作を記述する入出力制御デバイスを用いたユビキタスコンピューティングを提案している [1]．提案する入出力制御デバイスは，機能をできる限りシンプルに抑える一方，動作を動的に切り替えられる汎用性をもつ．

本稿では，このような入出力制御デバイス (ユビキタスコンピュータ) の動作記述言語の設計について述べる．提案するユビキタスコンピュータは他のコンピュータとの連携が必要であり，また，動的な動作変更が必要である．さらに，対象とするユビキタスコンピュータが小型・軽量・非力であるため，動作記述言語もできるだけシンプルで処理が簡単なものが望ましい．そこで，本研究ではイベント駆動型言語である ECA ルールの枠組みを用いてユビキタスコンピュータのための言語を設計する．コンピュータの機能をルールの集合として表現し，ルールの追加・削除による動的な機能変更を実現する．

2. ECA ルールの設計

ECA ルールとは，データベース技術の 1 つであるアクティブデータベースにおいて用いられている動作記述言語で，イベント・コンディション・アクションの 3 つを一組にして動作を記述する．イベントは環境内に起こる事象，コンディションはルールを実行するための制約条件，アクションは実行する操作を表す．システムの動作はこのようなルールの集合として表現されるため，ECA ルールを追加・削除することで自由に機能の変更が可能であり，ルールを部分的に変更することで柔軟なカスタマイズが行える．また，ルールを送受信することで，自分用のルールを携帯端末に移したり，場所に応じて必要なルールを受け取る位置・状況依存サービスが実現できる．

一般に，ECA ルールの言語仕様を設計するにあたっては，イベント・コンディション・アクションにそれぞれ何を記述できるようにするかをまず決定する必要がある．その後，それぞれの複数記述を許可するか，メタルールやルールのプライオリティの存在を認めるかといった点を決定する [2]．

想定するユビキタスコンピュータは図 1 に示すように，各種のセンサやボタンなどの入力状態を取得して処理を行い，さまざまなデバイスへの出力を行う機器である．内部状態を保持するためのレジスタをいくつか保持し，不揮発性メモリにルールを記憶する [3]．センサやボタンを接続するポートとは別に

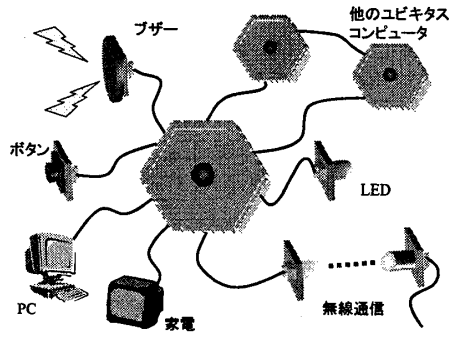

図 1: 想定するユビキタスコンピュータ

表 1: イベント

名称	内容
RECEIVE	シリアルポートからのデータパケット受信
TIMER	あらかじめ設定したタイマの発火

2 系統のシリアルポートを装備し，他のユビキタスコンピュータや PC，PDA などとメッセージ交換を行うことが可能である [4]．また，複数のシステムタイマをもち，ルールで自由に設定できる．さらに，ユビキタスコンピュータはルールによって自由に省電力モードに移行し，消費電力の節約を行なえる [5]．このようなユビキタスコンピュータの機能を実現するための言語仕様について，次節より詳細な設計を行う．

2.1 イベント

ルールの中で記述できるイベントを表 1 に示す．ユビキタスコンピュータに対する入力としてはシリアルポートからのパケット受信およびセンサなどをつないだポートからの信号がある．パケットに関しては受信したときに Receive イベントを発火させるが，ポートの信号に関してはそれぞれのポートが ON/OFF の状態を保持しているため，イベントとしては取り扱わずにコンディションとして扱う．そのため，ポートの状態のみでルールが発火させられるように，イベントをもたないルールを記述できるようにした．

2.2 コンディション

コンディションとしては，内部変数とポートの入力状態を記述できるようにした．複数の変数や複数の入力ポートの状態をまとめて記述でき，それらの条件がすべて満たされたときのみアクションが実行される．

2.3 アクション

ルールの中で記述できるアクションを表 2 に示す．OUTPUT アクションは各ポートの出力 ON/OFF を切り替え，出力ポートにつないだブザーや LED(Light Emitting Diode) などを制御する．OUTPUT_STATE アクションは，内部変数の値を変更する．TIMER_SET アクションは内部タイマを設定する．SEND_MESSAGE アクションは，特定の ID をもつメッセー

*1 : 大阪大学サイバーメディアセンター, Cybermedia Center, Osaka University
*2 : 大阪大学大学院情報科学研究科, Graduate School of Information Science and Technology, Osaka University
*3 : 大阪大学大学院工学研究科, Graduate School of Engineering, Osaka University(現在, 静岡大学情報学部)
*4 : NEC インターネットシステム研究所, Internet Systems Research Laboratories,NEC Corporation

表 2: アクション

名称	内容
OUTPUT	出力ポートの ON/OFF 設定
OUTPUT_STATE	状態変数の値設定
TIMER_SET	タイマの設定
SEND_MESSAGE	メッセージ送信
SEND_COMMAND	コマンド送信
HARDWARE_CONTROL	機器制御

表 3: コマンド一覧

名称	内容
ADD_ECA	ECA ルールの追加
DELETE_ECA	ECA ルールの削除
REQUEST_ECA	ECA ルールの要求

ジをシリアルポートを通して送信する．メッセージを受信したユビキタスコンピュータ上では RECEIVE イベントが発火するので，そのイベントに対する処理を書いておけばユビキタスコンピュータ間の連携が可能となる．SEND_COMMAND アクションは，他のユビキタスコンピュータに対して制御コマンドを送信する．制御コマンドとして扱える内容を表 3 に示す．制御コマンドはメッセージ送信と異なり，受信したコンピュータ側でイベントが発生することなくシステムで処理される．HARDWARE_CONTROL アクションは，ユビキタスコンピュータの機器制御を行ない，汎用 LED の ON/OFF/点滅制御や，省電力モードへの移行，シリアルポートのリレー制御等を実行する．1 つのルール中にアクションはいくつでも記述でき，記述した順番に実行されるものとする．

2.4 その他の設計項目

あるイベントが発火し，対象イベントを処理するルールが複数ある場合には，どのルールから処理するかによって結果が変わる場合がある．このため，ルールにプライオリティを表すパラメータを付加することがあるが，本研究では処理を簡単化するために，ルールにはプライオリティを設定せずに格納された順に評価するようにした．

また，ルールをグループ化することはルールを管理するにあたって有効であるが，本研究ではもともと大量のルールを格納することを想定しておらず，処理も複雑化することからグループの概念は取り入れていない．

3. ECA ルールのコーディング

ECA ルールはビット列に変換されてユビキタスコンピュータに格納される．ルールは 2 バイトをひとつの単位として記述され，図 2 に示す規則に沿ってコーディングされる．イベントは 2〜3 ビット目に記述し，コンディションは 4〜16 ビット目に記述するが，記述内容は図に示すようにイベントによって異なる．マルチコンディションフラグを 1 にした場合，次の 2 バイトにはアクションではなく 2 つ目のコンディションを記述する．アクションの先頭ビットは複数のアクションを記述するかを決定する継続フラグとなっており，継続フラグが 1 である限りアクションを続けて記述する．アクションは図に示すようにそのアクション内容によってフォーマットが異なる．まず 2 ビット目で Output/Output_State アクションとそれ以外のアクションを区別し，Output/Output_State アクションの場合は残りの 14 ビットで出力を指定する．その際，オフセットフラグを用いてポート 1〜6 と 7〜12 を切り替える．それ以外のアクションの場合はさらに先頭 3 ビットでアクションの種類を決定し，残り 11 ビットでそれぞれのアクション内容を記述する．

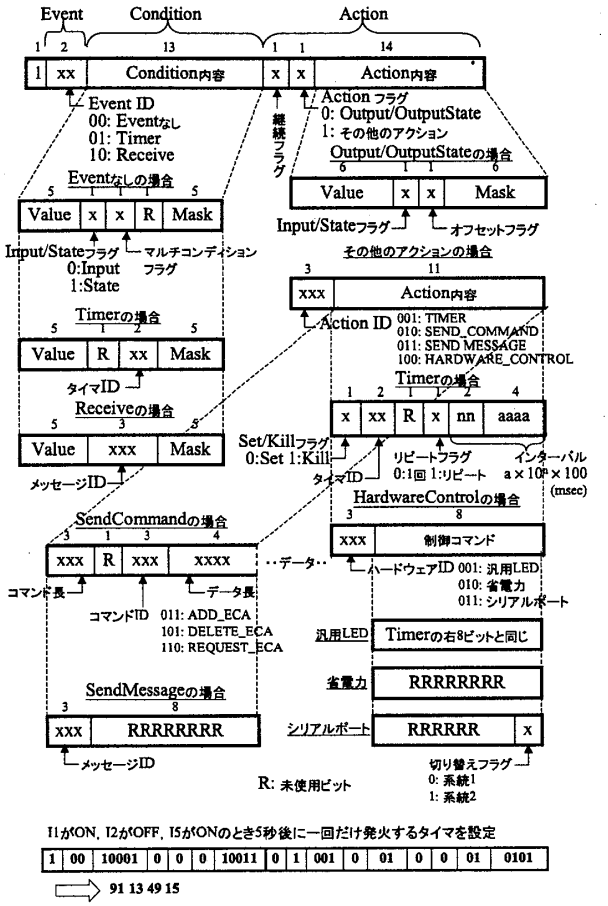

図 2: ECA ルールフォーマット

図 2 下に示すルール例は「91 13 49 15」の 4 バイトにコーディングされる．

4. 最後に

本稿では，ユビキタスコンピュータのための動作記述言語の設計について述べた．提案する言語は ECA ルールをもとにして設計しており，ルールの追加・削除による動的なシステム更新や，メッセージによる連携動作など柔軟なサービスが記述できる．今後はルールの無限連鎖といった ECA ルールの異常動作を検出するルールチェックシステムを構築する予定である．

参考文献

[1] 塚本ほか: ユビキタスコンピューティングを実現するためのルールに基づく入出力制御デバイス，FIT2002 論文集 (2002 掲載予定)．

[2] Widom, J. and Ceri, S.: 1.3 Rule Language Design, ACTIVE DATABASE SYSTEMS, Morgan Kaufmann Publishers, Inc. (1996).

[3] 早川ほか: ユビキタスコンピューティングのための入出力デバイスのハードウェアアーキテクチャ，FIT2002 論文集 (2002 掲載予定)．

[4] 岸野ほか: ユビキタスコンピューティングのための入出力制御デバイスの PC 統合環境，FIT2002 論文集 (2002 掲載予定)．

[5] 義久ほか: ユビキタスコンピューティングのための入出力制御デバイスのソフトウェアアーキテクチャ，FIT2002 論文集 (2002 掲載予定)．