

## 物理世界の状態変化に敏感なシステム PULSER-II の提案と実装\*

4 R-3

矢向 高弘†

菅原 智義‡

安西 祐一郎§

慶應義塾大学¶

## 1 はじめに

人間や自然の活動をもプロセスとして見る物理世界計算システム (PWCS: Physical World Computing System) を想定する。PWCS では、物理世界に直接働きかけられることが必要であるため、通信機能を持つ自律移動ロボットが従来のコンピュータの手足の役割を担うことが予想される。このように人間とロボットが同じ物理世界で共存する場合、物理的な共有資源のオーナーシップ問題が発生し、これを解決するためにはその一つの解である Personal-space-based Ownership Model (POM) をロボットの制御に応用する必要がある [1]。ところが POM を実装するには、物理世界の動作や状態の変化を敏感に感じ取り、反射的な動作を行なえるシステムが必要である。

本稿では、自律移動ロボットを制御するコンピュータ (以下ロボット・コンピュータと略す) に実装されている自律移動ロボット用 OS PULSER [3] 上に、物理世界の動作や状態の変化に敏感なシステム PULSER-II を実装する。

## 2 敏感なシステムの設計

物理世界では、コンピュータは資源を排他制御できない。またコンピュータは、人間や自然活動のパターン全てを想定できない。そのためロボット・コンピュータは常に物理世界を観測し、その時々状態に応じて適当な動作を行なうリアクティブな処理が要求される。

ロボットが対応すべき物理世界の状態は、センサから読みとられる値だけに限らず、過去の状態や他のロボットの状態など、様々な情報を処理した結果として得られる情報も含まれる。従って、単純にセンサの値の変化により割り込みを発生させるなどの実装方法では対応しきれない場合が多く、様々な処理を行なった後に得られる情報を状態として考え、その状態に応じた動作を実行する必要がある。

また、ある状態変化が起きた時、即座にプログラムに反映させなければリアクティブな動作を行なうことはできない。これには割り込み処理が適当であるが、従来の割り込み機構はプログラムに入れ子構造を強いるため、状態変化に応じた柔軟な状態遷移を行なうことはできない。しかし、我々が開発したマルチタスク、マルチスレッドの OS PULSER [3] が提供している Direct-Interrupt 機構は、入れ子構造を強いることなく割り込み機構と同様の即時性

を提供している。さらにユーザが Direct-Interrupt を自由に発生させることができるため、ハードウェアで提供されている割り込み機構よりも柔軟な利用法をとることができる。そこで、Direct-Interrupt により状態の変化を通知することにした。

しかし、Direct-Interrupt による通知だけでは不十分である。一般にバス・プランニングなどの推論を行なう場合、時折物理世界を観測し物理世界の内部表現を作り出し、これを用いて推論を行なうが、常にこの内部表現を物理世界の状態と一致させておくのは困難である。そこで新たに動的変数 (Dynamic Variable) を導入する。動的変数とは、あらかじめ物理世界の内部表現として計算式を定義しておくことにより、その値が常に現実の物理世界の状態と一致しているようにシステム側で値を維持している変数である。

Direct-Interrupt による物理世界の状態変化の通知機能と、動的変数による内部表現の維持機能により、物理世界の状態変化に敏感に反応できるプログラムが容易に記述できる。

## 3 敏感なシステム PULSER-II

PULSER-II は、プログラムが物理世界の状態変化に敏感に反応できるように、マルチタスク、マルチスレッド [2] の OS PULSER に動的変数を追加したシステムである。PULSER-II の監視スレッドによる動的変数の更新方法と、Direct-Interrupt を用いた状態変化の通知方法を以下に示す。

## Direct-Interrupt による通知

PULSER では、ユーザが定義した条件により Direct-Interrupt を発生する機構として、監視スレッド (inspecting thread) を導入している。これはシステムに一つ存在するスレッドで、ユーザが定義した条件全てを逐次検査し、条件が成立する時だけあらかじめ指定された Direct-Interrupt を発生させる機構である。この機構は、リアルタイム OSなどで複数のタスクがそれぞれ条件式を検査するスレッドを用意する場合に比べて、スレッド数が減り、コンテキスト切替の回数も減るなど、オーバーヘッドの削減に貢献している。

## 動的変数の実現法

動的変数の値を更新するにも、監視スレッドを用いた。動的変数を監視スレッドとユーザタスクの共有メモリに配置し、監視スレッドがユーザタスクの変数を直接変更することにより実現した。

\*Physical World Sensitive System PULSER-II

†Takahiro YAKOH

‡Tomoyoshi SUGAWARA

§Yuichiro ANZAI

¶Keio University

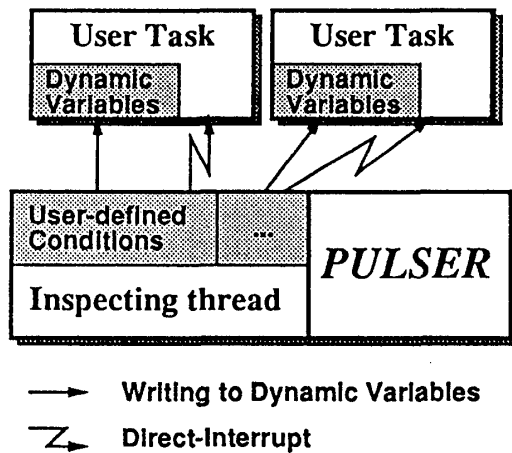


図 1: PULSER-IIにおける動的変数の更新方法、および状態変化の通知方法

以上の二つの機能の概念図を図 1 に示す。これらの機能により、物理世界の状態変化に対して敏感に反応することができる。

図 2 に C 言語のサンプルプログラムを示す。図 2 上のプログラムは監視スレッドが実行し、プログラムの値が TRUE の時に指定された Direct-Interrupt INTR\_NO を発生する。

図 2 下のプログラムはユーザタスクが実行する。関数 `intr_set()` により上のプログラムが監視スレッドに渡される。通常は `main_process()` を実行するが、Direct-Interrupt が発生した場合は `intr_handler()` が起動され、Direct-Interrupt がハンドリングされる。

変数 `dist` は動的変数であり、監視スレッドが動作する度にセンサの値を代入しているため、ユーザタスクでは常に最新のセンサの値として使用することができる。

実際には、PULSER にはマルチスレッドプログラム構成支援ツール MTP[4] が提供されているため、この例よりも簡潔に記述することが可能である。

#### 4 まとめと展望

本稿では、物理世界の動作や状態の変化に敏感なシステム PULSER-II を提案し、実装した。物理世界の状態を内部表現にするには、画像処理のように大きな計算量を必要とする場合が多く、目的とするタスクの速度低下を招く恐れがある。しかし我々の実装法では、内部表現に変換する監視スレッドと利用するユーザタスクが完全に切り分けられているため、マルチプロセッサなどを用いることにより容易に並列処理を行ない、高速化を図ることが可能である。

プログラム理論における動的変数のセマンティクスは未定であり、効果的な使用法とともに今後検討すべき課題である。応用の仕方にもよるが、動的変数は既存のシステムを容易にアクティブなシステムへと移行させることができるのではないかと考えている。

```
extern dynamic int dist;
intr_condition(){
    if((dist = sensor_read(FORWARD)) < 20){
        return(TRUE);
    } else {
        return(FALSE);
    }
}
```

```
dynamic int dist;
intr_handler(){
    while(TRUE){
        release();
        /* interrupt is handled here */
    }
}

main(){
    th=thread_create();
    type.type = INTR;
    type.intr = INTR_NO;
    thread_start(th, intr_handler, &type);
    intr_set(intr_condition, INTR_NO);

    main_process(); /* main routine */
    return;
}
```

図 2: 動的変数と監視スレッドのサンプルプログラム

#### 参考文献

- [1] 矢向 高弘、安西 祐一郎: 物理的な共有資源のオーナーシップ問題に関する考察, 日本ソフトウェア科学会第 8 回大会論文集, 45-48, 1991.
- [2] A. Tevanian, R. Rashid, D. Golub, D. Black, E. Cooper and M. Young: Mach threads and the UNIX kernel: The battle for control, Proceedings of the Summer USENIX Conference, 185-197, 1987.
- [3] 菅原 智義、矢向 高弘、安西 祐一郎: 自律移動ロボット用 OS PULSER の設計と実装, 情報処理学会第 44 回 (平成 4 年前期) 全国大会予稿集, 1992.
- [4] 紺田 和宣、矢向 高弘、安西 祐一郎: マルチスレッドプログラム構成支援ツール MTP の設計と実装, 情報処理学会第 44 回 (平成 4 年前期) 全国大会予稿集, 1992.