

UCF メッセージによる柔軟な機器ソフトウェア管理方法

○岡田 卓也* 平中 幸雄 武田 利浩

山形大学大学院理工学研究科

1. 序論

1.1 研究背景

近年、コンピュータネットワークは急速に私達の社会に浸透し、目覚ましい発展と進歩を続けている。今もネットワークは、内部を流れる情報量を増やし、情報処理を行う機器も高速な物へと、変遷を続けている。そうしたネットワークの研究において、センサネットワークの研究が近年注目を受けるようになってきている。

これまでのセンサネットワークでは、センサーノードと呼ばれるある特定の情報を取得する機器を、広範囲に配置し、その周囲一帯の情報をネットワークによって収集し、活用するものである。測定する情報や、情報を活用する分野など、様々な応用が期待されている。

しかし、使用する大部分のセンサーノードは、その演算能力や測定装置、バッテリーといった、いわゆる機器資源に乏しいものがほとんどである。こういった資源が乏しい機器では、新しい機能の追加や、変更、更新を行う際、その条件を満たすことが困難になる。また、物理的に機器を拡張し、要求を満たすようにしても、センサネットワークでは数多くの機器を用いているため、コストがかかってしまう。

1.2 現状の機器ソフトウェアが持つ課題

現状の機器ソフトウェアは独立して動作ができるよう設計されている。その設計内容は、あくまで機器ソフトウェア内で想定されている動作を行うためだけのものであり、それ以外の動作はできないものとして設計をされている。つまり、そのソフトにおいてできること、できないことが明確に分けられている。また機器毎にも、ハード面やソフト面においてメモリ量や処理速度、プログラムデータサイズなどが予め決定されており、それ以外の様々な機能の導入ということには、制約がある。

1.3 問題の解決に向け

予め決められた動作を行うものから、それ以外の動作に関しても実現できるような環境とするには、お互いの内容を理解する必要がある。例えば「メモリの消費量が大きく負荷がかかってしまうようなソフトがあり、全体に影響が出ている」といった状態を解決するには、メモリを消費しているソフトと、メモリを管理するソフトが情報をやり取りし、解決に向け動作を選択することで解決が望める。つまり、お互いが相互に連携を取り合い、問題に対処する枠組みがあれば、この課題は解決できる。またこの課題が解決できれば、センサネットワーク等の活用に幅が生まれる。

1.4 研究目的・方法

本研究は、組み込み系機器に搭載されたソフトウェアをこれまでの管理方法より、柔軟に管理させる方法を策定する。その方法として、機器上にソフトウェアを構築し、そのソフトウェアを柔軟に管理させる。この柔軟な管理には、相互通信が必須であるため、本研究ではこの相互通信に UCF を用いている。UCF メッセージを用いてソフトウェアを管理する枠組みを設計・作成し、その柔軟性を検証する。

2. UCF の概要

現行のネットワーク通信は、様々な機器ごとに異なる通信フォーマットを使用しているのが一般的である。よって通信を行うには、同一フォーマット同士の通信か、もしくは双方に理解できるようフォーマットの変換が必要となる。しかしこの場合、存在する全てのフォーマットの変換は非現実的である。全てを変換せずに、理解できる中継機器を経由するといった方法も存在するが、直接的な解決ではなく、中継機器の存在が問題となる。この問題を解決するには、統一された汎用の通信フォーマットを用意してやればよい。

そこで全ての通信を表現する、汎用の通信フォーマットとして研究されているものが UCF(Universal Communication Format) である。UCF は全ての通信を表しており、現状の通信形態を行うすべての機器に搭載されるものとして研究されている。これが全ての機器に搭載されることで、これまでの通信で問題となっていた異なるフォーマットという概念が無くなるので、どのような機器相手でも、相互に通信することが可能となる。そのような環境を整えるためには、様々な機器において不具合の無い仕様としなければならない。全ての機器を扱えるフォーマットとすることが、UCF が持つ目標である。

2.1 UCF オブジェクト

UCF は、オブジェクト指向である。ここでいうオブジェクトとは、ある特定の機能を持つ実体を指す。本研究では、特定の機能を持たせたプログラムをオブジェクトとして扱うことにした。

UCF で記述したデータを使用し、通信を行うオブジェクトを UCF オブジェクトと呼ぶ。よって、UCF オブジェクトは UCF メッセージを解釈する機能が必要となる。これら UCF オブジェクトには、それぞれユニークな自身を表すオブジェクト名を持つ。このオブジェクト名と、メッセージ内のタグ名を比較することで、メッセージが自分宛に送られたものか判断する。

* 現在、日立東日本ソリューションズ

3. SunSPOT 概要

本研究で使用する機器には、無線通信を行う通信機器である、SunSPOT と呼ばれる小型センサーデバイスを用いることにした。これは Sun Microsystems 社が開発した小型組み込み系デバイスである。小型のセンサーデバイスとしてだけでなく、無線通信機能として ZigBee を有しており、無線センサーネットワークの構成が可能な機器である。独自の Java Virtual Machine を搭載しており、java でプログラムが可能で、従来の組み込み系機器の問題であった、組み込みにかかる手間の多さを解決している。

4. 柔軟な機器ソフトウェア管理

4.1 柔軟な機能とは

本研究で提案する柔軟性とは、通常であれば応えることができない要求も応答できるようにすることである。この、一見不可能な要求を、試行錯誤の結果可能な要求にする、この行動全般を選択可能とすることで、従来までの応答より、より柔軟な応答となり、様々な選択肢が増えることが期待できる。本研究では、この柔軟な対応ができる機能として、オブジェクト自動ロード・アンロード機能とオブジェクト移動機能を提案する。この2つの機能の応用によって、グリッドコンピューティングの負荷分散のような、ソフトウェア単位の負荷分散が可能となる。

4.2 提案する管理機能 (1) オブジェクト自動ロード機能

オブジェクト自動ロード・アンロード機能とは、命令がされた時点で、その機能を有するソフトウェアオブジェクトがその命令だけで自動的に使用可能状態にロード、または終了状態となるアンロードがなされる機能である。

本機能は、このロード・アンロードの動作を、UCF メッセージを送信するだけの半自動でロード状態、またアンロード状態へと状態を変化させる機能である。これにより以下2点が可能となる。

1. 管理者による自由な機器の活用が可能
2. 管理者がロード・アンロードの状態を把握可能

機能を利用するために、ロード・アンロードをさせるオブジェクト名 Name とロード・アンロードの2つのどちらかを表す LOAD, UNLOAD の情報を与えることで利用が出来るように作成した。新しく定義した UCF オブジェクトは<load>で、次の構成とした。

UCF メッセージ例：

```
<load><s>SunSPOT[0]</s>Name, LOAD or UNLOAD</load>
```

4.3 提案する管理機能 (2) オブジェクト移動機能

機器内に存在するソフトウェアオブジェクトを、他の機器上へ“動作を継続させた状態”で移動させる機能である。これにより以下の応用が可能になる。

1. 異なる機器上へ機能の入れ替えが可能
2. 機器のトラブルによる動作継続の問題を解決可能
3. 機器にかかる負荷の分散が可能

本研究では、ソフトウェアオブジェクトの移動を行わせる技術として、マイグレーションという機能に着目した。この技術は、広義の意味では、物理的な機器の移動をはじめ、データ、プログラムなどを環境が異なる場所に移動させることをさす。

SunSPOT のマイグレーションは、アイソレイトというプログラムの単位を他の SunSPOT 上へ、移動させるものである。よって本研究でも移動するオブジェクトをアイソレイトという単位で構成することにした。アイソレイトとは Java のプロセスに相当する単位であり、稼動する Virtual Machine (以降、VM と記す) 上に複数実行が可能なものである。他のアイソレイトとは独立状態にあり、あるアイソレイト上でプログラム上に変化が起こっても、基本的に分離され影響を受けることがない。

SunSPOT には、このアイソレイトの移動機能が搭載されていたため、この技術を応用しソフトウェアオブジェクトの移動機能を作成した。

機能を利用するためには、マイグレーションをさせるオブジェクト名 MIG_Name とマイグレーションする先の SunSPOT を表すアドレス DEST を必要とする。新しく定義した UCF オブジェクトは<mig>で、次の構成とした。

UCF メッセージ例：

```
<mig><s>SunSPOT[0]</s>MIG_Name, DEST</mig>
```

4.4 管理機能を応用した柔軟なロード動作

オブジェクトロード・アンロードオブジェクトである<load>と、オブジェクト移動オブジェクトの<mig>を組み合わせることで、提案する柔軟な機器管理が可能となる。組み合わせる手順は次のようになる。

1. 資源が限界状態の機器にオブジェクトをロードする要求が発生する。
2. 限界状態の機器は、周囲の機器に協力要請を送信する。
3. 要請を受けた機器が協力要請の応答を行う。
4. 協力要請の応答を受けた結果を集計し、もっとも条件の良い機器を選択する。
5. 選択した機器へ、限界状態の機器から、不要と判断されたソフトウェアを移動させる。
6. ソフトウェアを移動したことによって限界状態から回復し、1のロード要求を実行する。

上記のような動作を行うことで、図のような柔軟な機器管理が可能となる。

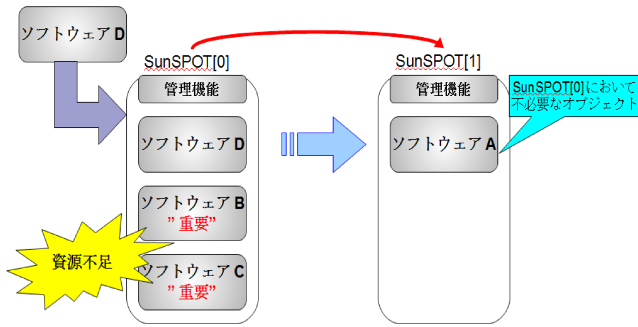


図1. 柔軟なロード例

上図は、2 台の SunSPOT 間で行う柔軟なロード例を表している。左の SunSPOT[0] がすでに資源不足になっている状態で、さらに新しいソフトウェア D をロードしようとしている。通常では、資源不足の為実現させることが出来ないが、上記手順で SunSPOT[0] は、周囲の機器に協力を要請する。すると要請を受けた SunSPOT[1] は、協力が可能である連絡を返信し、それを踏まえたうえで、SunSPOT[0] は既に実行しているソフトウェア群のうち、重要ではないソフトウェアを選別し SunSPOT[1] へ移動をさせている。その結果、SunSPOT[0] 内に資源の余裕が生まれ、当初の要求であったソフトウェア D のロードが可能になったことを示している。

4.5 柔軟な管理を受けるソフトウェアオブジェクト群

柔軟な管理を受けるソフトウェアオブジェクトとして今回は SunSPOT の搭載機能である各種センサを利用するソフトウェアオブジェクト等を作成した。

1. 温度センサ操作用オブジェクト
2. 照度センサ操作用オブジェクト
3. 加速度センサ操作用オブジェクト
4. LED 操作用オブジェクト
5. エコーオブジェクト
6. 負荷専用オブジェクト

各種オブジェクトは、SunSPOT 内に 5 つまでしか起動できないよう、制限を設けた。これは、擬似的に SunSPOT 機器の限界をあらわすためである。また各オブジェクトは、同一 SunSPOT 内に同時に 2 つ以上は起動しないようにも制限をかけている。以下はこれら各種オブジェクトの説明である。

5. 提案する管理方法の動作実験

5.1 GUI による管理支援ツールの作成

これまで作成した機能を視覚的に確認するため、GUI ツールを用いたソフトウェアを作成し動作実験を行った。この GUI ツールによる管理には以下の効果が期待できる。

1. 視覚的に変化が理解しやすい
2. 長文傾向にある UCF メッセージの入力支援

作成したツールには、管理者向けの機能として、UCF メッセージ入力支援部分、SunSPOT 状態表示部分、メッセージログ+メッセージレスポンス表示部分を設けた。このツールを用いて、柔軟な管理の動作実験を行った。

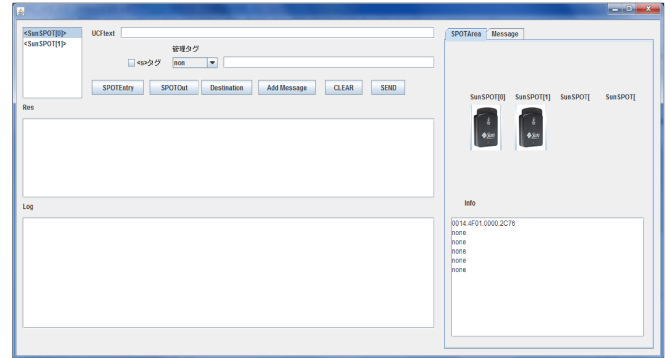


図2. 作成した GUI ツール画面

5.2 動作確認実験

柔軟な機器管理が正常に動作するか、以下の動作実験をおこなった。まず SunSPOT2 台を用いて、ソフトウェアの負荷分散動作を確認する。片方の SunSPOT に予め設定した機器の限界値（ソフトウェアの起動数、本実験では、起動数を 5 とした）までソフトウェアをロードしておき、その上で限界状態にある SunSPOT に新たにソフトウェアをロードさせる。ここでもう片方の SunSPOT と連携し、負荷分散を行い、柔軟にロードできるか動作確認を行った。その結果、意図した動作を確認できた。

5.3 柔軟な管理機能の応用事例

作成した柔軟な機器管理を応用することで、グリッドコンピューティングのような様々な効果が期待できる。

まず第 1 に、管理者が個々の機器性能を把握することを必要としなくなる。これはグリッドコンピューティングが持つ仮想化の特性を表している。ネットワーク上にある全ての機器を見かけ上 1 つの機器として扱うことになるので、詳細な状態の把握をする必要がなくなる。

そして、第 2 に管理者は扱う資源の物理的な場所や性能を予め知る必要がなくなる。多数の機器と連携・協力を行うことで資源の統合化が実現できる。

第 3 に機器の自立化が可能となる。機器それぞれが独立して動作し、各自で自分の状態を管理しながら、外部の状態やメッセージに大して適切な動作を取っており、環境内に新たに接続したプログラムやユーザといった外部状態の変化に対しても柔軟に対応が可能となる。これにより特有なエラーからの復帰に対処することが可能となる管理者の問題解決や、変化の全てに即座に対応することが可能となる。

6. 今後の課題

今回の研究によって、負荷の高い機器から、負荷の低い機器へ連携を行い、ソフトウェアの管理を柔軟に行わせる技術

がUCFメッセージによって構築できた。今後は、より柔軟な機能とするために必要な機能の追加、また組み込み機器に必要な技術の応用が求められる。本研究では、以下の4点について検証する。

- アイソレイトの選定機能
- 移動先の選定機能
- ソフトウェアの移動に関する問題
- リアルタイム性

6.1 アイソレイトの選定問題

稼働させるアイソレイトの選定基準をどのように設けるべきかという問題が考えられる。センサネットワークの運用を考慮するとデータ取得しているソフトウェアが移動してしまうと、必要な情報が得られなくなる可能性がある。このように、移動しても影響のないアイソレイトの選定条件の策定が必要になる。一般的に用いられる選定基準には、優先度やサイズなどがある。背景にあるセンサネットワークを考慮すると、センシングを行っているような、特有の場所でなければ動作が無意味となるようなソフトウェアは必然的に優先度は高くなる。このように、センサネットワークの運用部分にも対応できるよう、これらの条件を含んだ機能の追加が必要となる。

6.2 移動先の選定問題

センサネットワークのように、多くのノードを使用する中で、最も条件の良い機器を選定しなければならないが、条件によっては複数存在したり、まったく無い場合が出てくる。また選定の条件には、ノードの残り資源量や、バッテリー残量、物理的な距離などを考慮しなければならない。同一の条件を持つ機器が複数存在する場合は、固有のアドレスなどによる選定、条件に見合う機器が存在しない場合は、問題を発生させた機器内で解決できるような機能を追加するといった、機能の追加が必要であり、また選定条件も含んだ機能の追加が必要になる。

6.3 ソフトウェアの移動に関する問題

本研究で作成したオブジェクトの移動機能だが、バイトコードを持たない機器へ移動ができていない。使用したSunSPOTのVMであるSquawk Vmの仕様のため、解決できなかった。しかしリモートロードという技術があり、これを応用することで問題は解決するものと思われる。SunSPOTのVMはオープンソースであり、導入すること事態可能であることから、応用は十分に可能であると考えられる。

6.4 リアルタイム性

本研究では移動にマイグレーションという技術に応用したが、移動完了まで時間がかかっている。これでは、柔軟な管理だとしても組み込み機器としては致命的である。マイグレーションの高速化の技術としてはこれまでにLive Migrationという技術が紹介されており、この技術に応用することで高速化が実現できると思われる。

7. まとめ

本研究では、機器ソフトウェアの柔軟な管理を目的とした。実現方法としてUCFメッセージによる機器毎の負荷分散を提案した。内容は、機器内のソフトウェアを移動させることで負荷分散を行い、周囲の機器と連携し、不可能な要求に対して要求を可能とさせる枠組みを作成した。またより柔軟な管理とするために必要な機能の提案と応用すべき技術の調査を行った。

参 考 文 献

- [1] Li Chen, Phillip McKerrow, Qinghua Lu, “Developing Real-time Applications with Java Based Sun SPOT,” Faculty of Informatics Papers, 2008.
- [2] Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, “Live Migration of Virtual Machines” Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation, Volume 2, pp. 273-286, 2005.
- [3] Chandrasekhar Boyapati, Alexandru Salcianu, William Beebe, Jr., Martin Rinard, “Ownership Types for Safe Region-Based memory Management in Real-Time Java,” Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation, pp. 324-337, 2003.
- [4] Li Chen, Phillip McKerrow, Qinghua Lu, “Developing Real-time Applications with Java Based Sun SPOT,” Faculty of Informatics Papers, 2008.
- [5] 福沢尚司, 宋 国煥, 富樫 敦, 白鳥則郎, “命題論理に基づいたソフトウェア設計支援システムとその応用,” 情報処理学会研究報告, マルチメディア通信と分散処理研究会報告, マルチメディア通信と分散処理, 76-1, 1996.
- [6] 鄭顕志, 末永俊一郎, 中村善行, 清雄一, 吉岡信和, 深澤良彰, 本位田真一, “XACプロジェクト オープンセンサネットワークミドルウェア研究開発プロジェクト,” CENTER FOR GLOBAL RESEARCH IN ADVANCED SOFTWARE SCIENCE AND ENGINEERING NATIONAL INSTITUTE OF INFORMATICS, GRACE TECHNICAL REPORTS, 2009.
- [7] 平中幸雄, “UCF Universal Communication Format 汎用通信フォーマット,” 2008.
- [8] 渡邊正裕, “UCFを使用する基本通信オブジェクトの開発,” 山形大学工学部情報科学科, 平中研究室, 平成19年度卒業論文.
- [9] 岡田卓也, 那須弘人, 武田利浩, 平中幸雄, “UCFメッセージによる柔軟な機器ソフトウェア管理方法,” 山形大学工学部情報科学科, 平中研究室, 平成21年度卒業論文.