

C-020

## UCF メッセージによる柔軟な機器ソフトウェア管理方法 A Flexible Software Management Using UCF Messages

岡田 卓也<sup>†</sup> 那須 弘人<sup>‡</sup> 武田 利浩<sup>†</sup> 平中 幸雄<sup>†</sup>

Takuya Okada Nasu Hiroto Toshihiro Taketa Yukio Hiranaka

### 1. はじめに

私たちの身の回りには様々な機器ソフトウェアが存在している。携帯電話をはじめ、多種多様な機能を持つ便利な機器があふれる中、いまだ多くの課題も残している。それは例えば、設計された時点で想定をされていない問題に直面した場合、多くの機器ではこの問題を自力で解決できず、多くのソフトウェアは対応できない。本研究では、こういった現状の機器ソフトウェアが持つ問題を UCF (Universal Communication Format)[1] を導入し、この UCF のメッセージで活用することで、今までにない柔軟な機器ソフトウェアの管理として解決を試みる。

### 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いる。さらに新たな機能の導入を行うには、機器で利用可能な通信形式などを使用しなければならないといった制約がかかってしまう。

そこで本研究では、ハード・ソフト両面で活用できるメッセージとして研究中のフォーマットである UCF を導入し、これらの解決を目指した。

### 3. UCF メッセージによる管理

UCF とは、汎用的に通信を表すことのできる統一フォーマットとして考案されたものである。フォーマットを統一することで、異なるフォーマットや、異なるレイヤー間を通信するクロスレイヤー通信も可能としている。本研究ではこの UCF フォーマットを用いたメッセージである UCF メッセージを用いることで、ハード・ソフト両面で相互通信を可能とし、レイヤーを越えた柔軟性の高い管理を行い、柔軟性の評価と、より詳細な管理方法について探索している。

### 4. 機器ソフトウェア構築

本研究では、構築する機器ソフトウェアに Sun Microsystems が開発した小型センサーデバイスユニットである SunSPOT を用いた。この機器は、Java ベースの VM

を搭載しているため、組み込み機器の中でも拡張性が高いものである。本機器ソフトウェア構成は、この SunSPOT のうち PC 接続用のユニットを経由し、実際の管理対象ソフトウェアが構築されたユニットへメッセージを送信し、その挙動を管理するものとした。

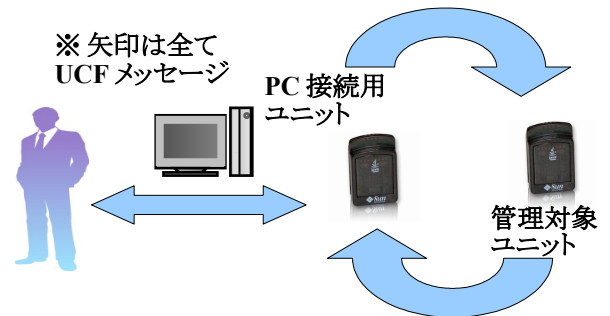


図1. ソフトウェア構成図

### 5. 機器プログラムに考慮すべき機能と実現方法

作成する機器ソフトウェアは、プログラム自動ロード・アンロード、メモリ管理を柔軟に行うよう構築した。これら機能により柔軟な管理が可能となる。機能の説明は以下の通りである

#### ● プログラム自動ロード・アンロード

プログラム自動ロード・アンロード機能とは、ソフトウェアを要求した時点で、その機能を有するプログラムが自動で起動または、終了がなされる機能である。この機能により、機器自身が持ち合わせていない機能を司るソフトウェアの導入が容易になるといった活用が可能となる。

またこの機能は、プログラムを異なる機器上に移動させる機能を持たせることで、より柔軟なプログラムの操作を行える。

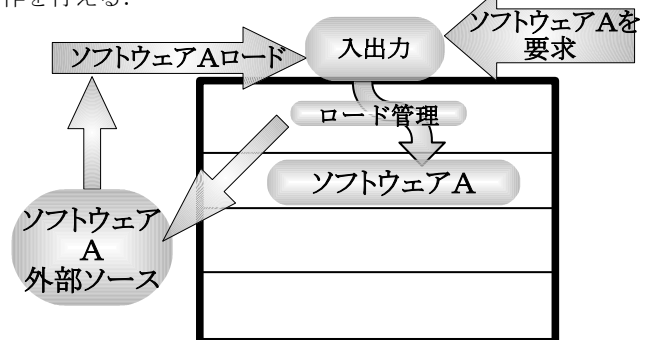


図2. ソフトウェアロードイメージ

<sup>†</sup> 山形大学 Yamagata University

<sup>‡</sup> 株式会社ハイテックシステム High Tech System

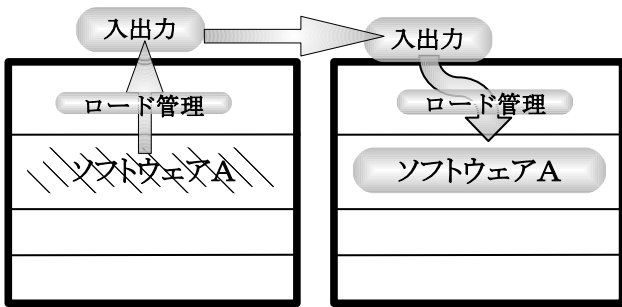


図3. ソフトウェア移動ロードイメージ

● メモリ管理機能

メモリ管理機能には、使用中のメモリの開放、確保を各ソフトウェア毎に交渉させ、割り当て、開放といった一連のメモリ操作を行わせる機能である。新たに追加させたい機能が、現状のメモリの状態により使用できない場合や、メモリの確保を任意に行いたい場合などに使用する。例えば、動作中のソフトにメモリを開放させて、メモリに容量を作り、空いたメモリを使用して、動作させるといった柔軟な対応が可能となる。下図では、この応用例として、メモリ使用率が著しいソフトAに交渉を行い、新しく導入するソフトD分のメモリを提供させるイメージを表している。

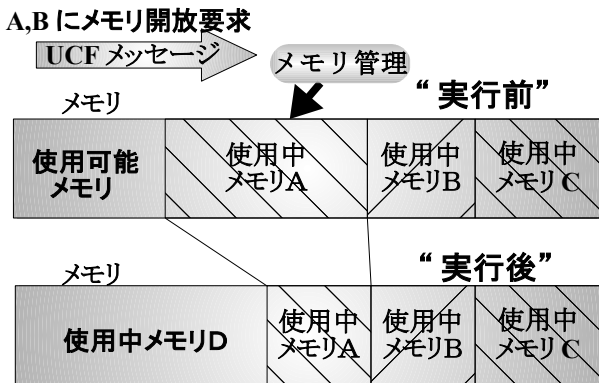


図4. メモリ管理イメージ

6. 実現できた柔軟な操作

● プログラム自動ロード・アンロード

ユーザーが任意で、プログラム動作状況を自由に変更することでより柔軟な選択が行えるよう実装した。管理するソフトウェアをUCFで表現するオブジェクト単位とし、このオブジェクトを定義することで、プログラムとして利用可能にし、プログラムのロード、定義をはずすことでアンロードの機能を実装させた。

今後は、外部ソースからのロード、異なる実機へのロード、アンロードなどの進んだロード処理を目指す。

● メモリ操作機能

ユーザーが任意のタイミングにて、全てのソフトウェアで使用中のメモリの開放を促す処理をさせる機能が実

現した。しかし、ソフトウェア指定のメモリ操作にはまだ至っていない。今後より柔軟な機能とするには、この機能を特定のソフトウェア単位で行わせることや、各アプリケーション同士で行うことを可能とさせる、といった工夫が必要となる。

7. 検討中の柔軟な機能例

● CPU管理機能

VMが行っているCPU管理機能を制御する機能。優先順位 (Priority) の決定などの処理を行う。CPUにおいて優先順位の変更などは変更が容易なので機能実現は容易であると思われる。

● 実行するプログラムの機器間の移動

稼動中のSunSPOT内において、メモリ不足やCPU負荷の増大といった、ハード面に起因するアプリケーションの動作に支障をきたす状態において、アプリケーションの負荷を軽減させるための機能として、稼動中のSunSPOT以外の機器にて動作をさせる機能を作成中である。これにより、ハード面の問題を自己のソフトウェア内で解決できない場合でも動作を行うことが可能となる。

8. 考察

機器ソフトウェアの機能をUCFオブジェクトとし、UCFメッセージを用いて管理する本手法は、オブジェクト操作として、直感的に理解がしやすく、操作の容易さが垣間見ることができる。また本来、ハードが持つ機能とソフトウェアとで直接通信を行い、レイヤー間通信を容易にするこのUCFを用いた手法は、これまでにない機器ソフトウェアの使用の可能性を開いたといえる。しかし柔軟な管理という点では、発生した問題を回避する方法が未だ乏しく、結果柔軟と呼ぶには未だ不十分であるといえる。これは、問題の回避方法をハード、ソフトの両面から検討することで解決できると思われる。少なくともUCFが持つ、汎用性を生かしながらの通信を行う事自体は実現できたので、今後はそれに見合う応答を、オブジェクトごとに検討することが必要であると思われる。

参考文献

[1] 渡邊 正裕, 田名部 創大, 武田 利浩, 平中 幸雄, “UCFを使用する基本通信オブジェクトの開発”, FIT2008(第7回情報科学技術フォーラム), 第4分冊 167-168, L-035 (2008).  
 [2] 平中 幸雄, 伊藤 大視, 市川 直樹, 武田 利浩, “イーサネットフレームによるダイレクトUCF通信”, FIT2007(第6回情報科学技術フォーラム), 第4分冊 23-24, L-011(2007).  
 [3] 渡部 良平, 渡邊 高成, 武田 利浩, 平中 幸雄, “通信路を問わない汎用通信フォーマットの提案と何でもリモコンの作成”, (第5回情報科学技術フォーラム), 第4分冊 81-82, L-033(2006).  
 [4] 平中 幸雄, 菅井 栄治, 渡部 修平, 武田 利浩, “汎用通信フォーマットによる移動型オブジェクトの実現”, (第4回情報科学技術フォーラム), 第4分冊 293-294, M-038(2005).  
 [5] Yukio Hiranaka, Hitoshi Sakakibara, Toshihiro Taketa, “Universal Communication Format for Multimedia Data”, Proc. Sixth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2005), 338-339 (2005).