

WebCodget : Web サーバに移動して動作する 組み込み機器向け移動ソフトウェア

河口 信夫[†] 春原 雅志[‡]

[†]名古屋大学情報連携基盤センター/情報科学研究科 〒464-8601 名古屋市千種区不老町 1

[‡]名古屋大学工学部 〒464-8601 名古屋市千種区不老町 1

E-mail: [†]kawaguti@nagoya-u.jp, [‡]sunohara@el.itc.nagoya-u.ac.jp

あらまし 近年、ネットワークに接続される組み込み機器においては、Web サーバを内蔵し、設定等を行う機器が増加している。利用者に特別な機器やソフトウェアが不要な点や、機器に複雑なユーザインタフェースを必要としないことから、ブロードバンドルータや無線 LAN アクセスポイントなどで、広く採用されている。しかし、多数の機器がネットワークに接続されるユビキタス環境においては、すべての機器に Web サーバが内蔵されていることは、むしろ弊害を生むことがある。本稿では、Web サーバを保持しなくとも、Web サーバに移動し、その上で動作するソフトウェアを保持することによって、同等の機能を持つ仕組みを提案する。各組み込み機器は、Web サーバを保持する必要がなく、また、Web サーバと機器の間は TCP/IP での接続を必要としないため、低コストでの実現が可能になる。さらに、1 台の Web サーバですべての機器をモニタリングすることが可能になる点や、動的な変更への対応が可能な点などの特徴を持つ。本稿では、基盤ソフト cogma 上に WebCodget として実装したので報告する。

キーワード Web サーバ, モバイルエージェント, 移動ソフトウェア, cogma, codget

WebCodget: Mobile Software which Migrate to Web Server for Embedded Devices

Nobuo KAWAGUCHI[†] and Masashi SUNOHARA[‡]

[†]Information Technology Center, Nagoya University 1 Furo-Cho, Chikusa-ku, Nagoya, 464-8601 Japan

[‡]Department of Engineering, Nagoya University 1 Furo-Cho, Chikusa-ku, Nagoya, 464-8601 Japan

E-mail: [†]kawaguti@nagoya-u.jp, [‡]sunohara@el.itc.nagoya-u.ac.jp

Abstract Recent years, networked embedded devices that have built-in web servers from where it can be used for setting up. Users don't have to use special devices or applications, and the simplicity of the devices user interface made it widely used in devices such as broadband routers and wireless access points. However, in ubiquitous environments where many devices are connected to the network, integrating web servers in every device can even have bad effects. In this paper we propose a method where even without having a built-in web server, a software running on the device can migrate to the web server, therefore providing similar functionality. Since there is no need to integrate web servers in each device, and a TCP/IP connection between the device and the web server is not necessary, a low-cost implementation is possible. Additionally, all devices can be monitored with one web server, and the possibility of a response to dynamic change is two characteristics. This paper also presents an implementation named "WebCodget" using "cogma" platform.

Keyword web server, mobile agent, mobile software, cogma, codget

1. はじめに

近年、ネットワークに接続される組み込み機器においては、Web サーバを内蔵し、機器の設定等を行う機器が増加している。利用者に特別な機器やソフトウェアが不要な点や、機器に複雑なユーザインタフェースを必要としないことから、ブロードバンドルータや無線 LAN アクセスポイントなどで、広く採用されている。しかし、多数の機器がネットワークに接続されるユビ

キタス環境においては、すべての機器に Web サーバが内蔵されていることは、むしろ弊害を生むことがある。例えば、センサデータを提供する機器では、個々のデータを取得するために、個々の Web サーバにアクセスする必要がある。センサが多数になると、アクセスすべき Web サーバの数が増え、管理が煩雑になる。他にも、IP ネットワーク上に Web サーバが増えることにより、ネットワークの管理コストも増加する。

本稿では、Web サーバを保持しなくとも、Web サーバに移動し、その上で動作するソフトウェアを保持することによって、同等の機能を持つ仕組みを提案する。具体的には、組み込み機器上には Web サーバではなく、ネットワーク上に存在する Web サーバ上で動作すべきソフトウェアを保持する。各組み込み機器は、Web サーバを保持する必要がなく、また、Web サーバと機器の間は TCP/IP での接続を必要としないため、低コストでの実現が可能になる。さらに、1 台の Web サーバですべての機器をモニタリングすることが可能になる点や、動的な変更への対応が可能となる点などの特徴を持つ。我々は、すでにユビキタスコンピューティング環境において、動的に変化する状況に柔軟に対応可能な基盤ソフトウェアとして“cogma”[1]の開発を進めてきた。我々は、本稿で提案する仕組みを cogma 上のアプリケーション WebCodget として実装した。

また、cogma は異機種間での移動ソフトウェアを実現するために Java 上で実装されている。一般に組み込み機器においては、Java はリソースを多く消費し、コストや速度の面から望ましくない。そこで、我々は cogma アーキテクチャを C 言語上でも動作可能とするために仮想 CPU を利用した VPcogma(Virtual Processor cogma)[2,3]を開発してきた。さらに、cogma との親和性を高めるために、Java から C 言語へのトランスレータを開発した。これにより、開発環境が充実し、開発効率が高い Java で開発を行い、C 言語へトランスレートして、リソースの制限されている組み込み機器上での動作を実現することができる。WebCodget もトランスレータによって Java で開発し、組み込み機器上で動作させることができる。

2. ユビキタス環境の基盤ソフト：cogma

我々は基盤ソフトウェアシステム cogma[1] (Cooperative Gadget for Mobile Appliances) を構築してきた。cogma は以下の特徴を持つ。

- A) 軽量ミドルウェア
Personal Java/ J2ME による省メモリで軽い実装
- B) ソフトウェアの動的な移送に対応
動作状況を保持したまま、端末間を移動可能
- C) 複数の異種ネットワークの同時利用が可能
TCP/IP, シリアル等の通信デバイスを利用可能
- D) 他ノードの自律的な発見機構
Hello Packet 等の利用による発見
- E) 移動ソフトウェアの管理・通信機構

cogma では、移動ソフトウェアを設計の基本としているため、ソフトウェアを必要に応じてインストールすることが可能になる。本稿でも、Web サーバ上に移動するソフトウェアの実装に、移動ソフトウェアを利

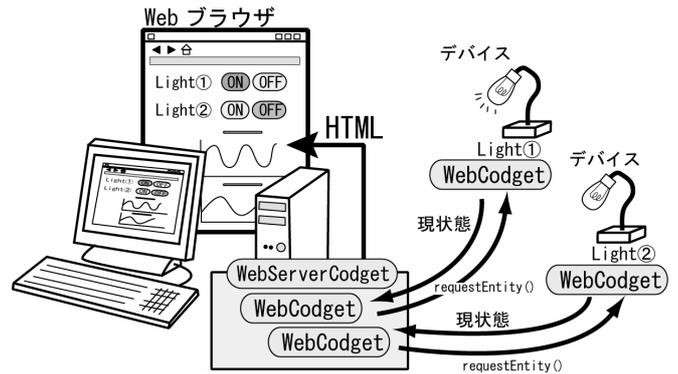


図 1 : WebCodget の概念図

用している。また、ネットワーク上に存在する Web サーバの発見にも、cogma のノード発見機構と、その上に構築された組み込み機器向けのサービス発見機構を利用している。

3. WebCodget の必要性と設計

3.1. 必要性

近年、ネットワーク組み込み機器では、Web サーバを搭載することにより、ユーザインタフェースの実装を簡易化することが多く行われている。組み込み Web サーバを利用している機器は、多くがブロードバンドルータや、無線 LAN アクセスポイントといった、比較的高い機能を持った機器である。しかし、ユビキタス社会が広がるにつれ、より低機能な機器もネットワークに接続されて利用されることが期待されている。これまでと同様に、個別の機器に Web サーバを搭載する仕組みをこのまま続けていくと、家庭やオフィスの中に、多数の Web サーバが設置され、ユーザは機器の設定や、状況を監視するために、多数の Web サーバを利用することが必要になる。本稿では、このような無駄な Web サーバの設置を避ける手法として、各組み込み機器には、Web ページを生成するソフトウェアのみを搭載し、必要に応じてネットワーク上の Web サーバに移動させることにより、機器の集中管理を実現する手法を提案する。本稿では、これを WebCodget と呼ぶ。

3.2. 概念

図 1 に WebCodget の概念図を示す。例えば、卓上ライトといった単純なデバイスに Web サーバが搭載されると、個々のライトにネットワークアドレスを付与し、Web ブラウザで直接アクセスする必要が生じる。一方 WebCodget の概念を用いると、個々のライトには、単純なライトとの通信と、Web ページの生成を受け持つ WebCodget と呼ばれる移動ソフトウェアが搭載され、ネットワーク上の Web サーバに移動する。ユーザは、1 台の Web サーバを見るだけで、複数のライトや、その他のセンサ情報などを確認することができる。ここ

で、WebCodget の利点として、各ライトデバイスは、Web サーバとの間で通信可能であれば良く、TCP/IP を解釈する必要はない。また、通信速度も制御のみの場合は、高速である必要はない。例えば、低速な電灯線通信（PLC）などを利用して、移動ソフトウェアを送ることも可能である。ソフトウェアの移送は初めてネットワークに接続したときのみであるため、転送時間は、ほとんど無視できる。これにより、ライトをコンセントに接続するだけで、Web サーバ上に部屋のすべてのライトの制御用のページが表示されることになり、ユーザの利便性が高まる。さらに、Web サーバ側でライトの WebCodget が生成する Web ページの位置やレイアウトなどを整理することにより、機器の管理も容易になる。

3.3. 要求仕様

では、このような WebCodget はどのように設計すべきかを検討する。以下に WebCodget が満たすべき要件を挙げる。

- 1 . Web サーバを発見する機構を持つこと
- 2 . Web サーバ上に移送可能なソフトウェア (WebCodget)を持つこと
- 3 . Web サーバ上に移送した WebCodget と通信し、機器の制御や情報伝達を行う機能を持つこと
また、WebCodget は Web サーバの存在なしには、実現できない。以下に Web サーバ側が満たすべき要件を挙げる。

- 1 . 他機器から発見される機構を持つこと
- 2 . 機器から移送されてくるソフトウェア (WebCodget)を実行すること
- 3 . ユーザからの Web ページリクエストを各 WebCodget に与え、ユーザに回答すること
- 4 . WebCodget を適切な Web ページに対応させること
- 5 . Web サーバ上で他の WebCodget と連携を実現可能とすること

上記のように、Web サーバ側にも WebCodget のための機能を持つことが期待されている。

4. cogma 上での実装

3 節で行った要求仕様の設計に従い、実装を行った。cogma 上には、ネットワーク上のサービスを発見する機構として、EmbeddedManager の仕組みが存在する。本実装では、Web サーバを発見するために、EmbeddedManager を利用した。Web サーバは cogma 上のアプリケーション (codget) として実装した。Web サーバは自身を EmbeddedManager に対し、WebCodget のためのサービスとして登録を行う。これにより、WebCodget は EmbeddedManager を通じて、Web サーバ

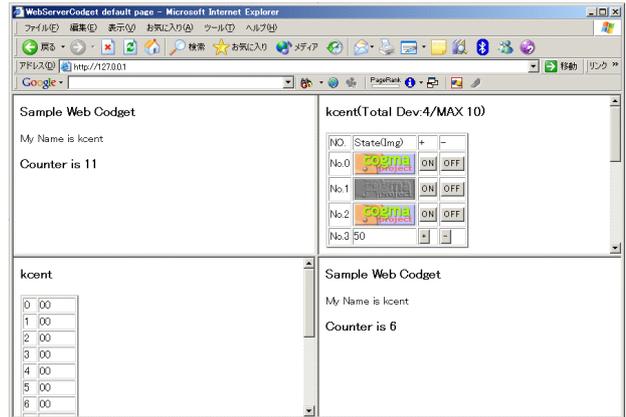


図 2 : Web ブラウザからみた WebCodget 群が動作するノードを発見することができる。Web サーバを発見した WebCodget は Web サーバ上にコピーを移動させる。これにより、機器上の WebCodget と、Web サーバ上の WebCodget が通信を行って、制御等の情報交換を行う。WebCodget は Web サーバ上に移送された後に、Web サーバ内に登録され、起動される。これにより、動的な Web ページの生成が実現できる。ユーザからの Web ページのリクエストを各 WebCodget に転送することにより、Web ページ上で機器からの情報確認や制御が実現できる。各 WebCodget は直接アクセス用のディレクトリ (ID) を与えられる。このディレクトリにアクセスすることにより、機器が生成するページを直接操作することもできる。また、Web サーバ側により適切に変換することも可能である。今回の実装では、Web ページの生成に ApacheJakarta プロジェクトの ECS(Element Construction Set)ライブラリを利用している。ライブラリをユーザ端末に合わせて動的に変更することにより、各 WebCodget 機器が持つソフトウェアは単純なままで高度な機能が実現できる。

図 2 に Web ブラウザにより表示された WebCodget を示す。ここでは 3 種類、4 個の WebCodget が動作している。現在の実装では、フレームを用いて、ページを分割しているが、今後はより洗練されたデザインを用いることを予定している。



図 3 : Vpcogma の動作環境

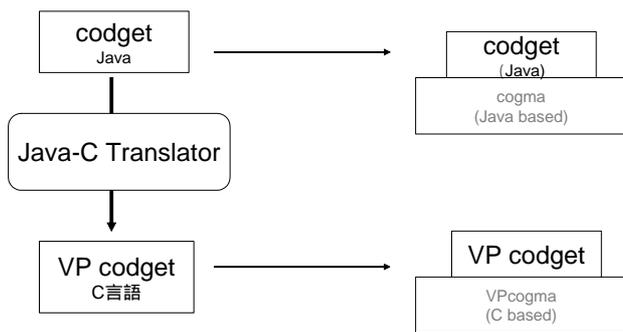


図 4 : Java C トランスレータの概念図

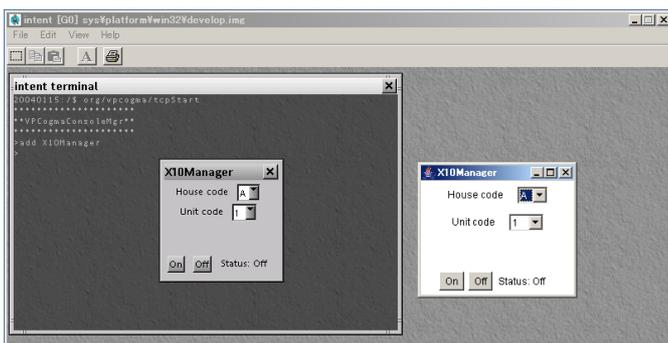


図 5 : C(左)と Java(右)の codget

5. VPcogma と Java C トランスレータ

cogma は、Java を用いて実装されているため、省リソースの組み込み機器には向かない。そこで、intent 上で実装されている仮想プロセッサの機構を用いることにより、C 言語での実装を行った。これを VPcogma と呼ぶ。図 3 に示すように、VPcogma は様々な OS や CPU を持った機器上で動作することができる。

さらに、VPcogma 上でのソフトウェアの開発を円滑に行うために、Java から C へのトランスレータを実装した。これにより、Java 上で実装された codget を VPcogma 上で簡単に動作させることができる。概念図を図 4 に、変換した codget の動作の様子を図 5 に示す。これにより、組み込み機器においても移動ソフトウェアを活用することが可能になった。

また、VPcogma とトランスレータの組み合わせにより開発効率も劇的に向上する。WebCodget はそもそも移動ソフトウェアであるため、ターゲット上に転送する際にも、開発ホストから移動させるだけで良い。そこで、本フレームワークを利用した WebCodget 開発は以下のような手順をとる。

- 1 . Java を用いてホスト上で WebCodget を開発
- 2 . cogma 上のネットワークエミュレータを用いてホスト上で動作確認・デバッグ
- 3 . 複数機器を用いて動作確認・デバッグ
各機器への WebCodget の配布も動的に可能
- 4 . Java->C トランスレータで VP 版 WebCodget を構築

5 . VPcogma 動作機器へ動的に WebCodget を 移送して、動作確認・デバッグ

上記のステップを経ることで、基本的なネットワーク通信機能は Java の豊富な開発ツールを利用して確認・デバッグできる。

6. まとめ

本稿では、Web サーバ上に移動して組み込み機器の制御を実現する WebCodget を提案した。WebCodget の利点を以下に挙げる。

- 1 . 各機器が Web サーバを保持しないため、
 - A. 機器が必要とするリソースが少なく
低コスト化が実現可能
 - B. ユーザ端末と直接 TCP/IP で接続する必要がないため、Web サーバ間との通信が柔軟
 - C. ユーザは多数の Web サーバを扱う必要がないため、管理・運用コストが低い
- 2 . Web サーバにソフトウェアが移動するため、
 - A. 機器の電源が切れていても動作することができる。これにより、各機器に永続的なメモリや電池などを持つ必要がなくなる
 - B. Web サーバ上で他の機器との連携を実現することができる
 - C. Web サーバを高度化することにより、WebCodget からの情報を、利用する端末に合わせた表示方式が実現できる。例えば、携帯電話からの操作を実現できる。
- 3 . Java C トランスレータの利用により、
 - A. 高い開発効率と、低コストを同時に実現
 今後の課題は、Java C トランスレータの完成度の向上や対応範囲の拡大。動作プラットフォームの拡大、様々なライブラリの構築などが挙げられる。また、本成果は名古屋大学込みソフトウェア技術者人材養成プログラム NEXCESS[5]にて活用されている。

本研究の一部は、文部科学省科学技術振興調整費（若手任期付研究員支援）の成果である。

文 献

- [1] Nobuo Kawaguchi, Cogma: A Middleware for Cooperative Smart Appliances for Ad hoc Environment, Proc. of International Conference on Mobile Computing and Ubiquitous Networking, ICMU2004, pp.146-151(2004).
- [2] 河口 信夫, 金岡 弘記, 木曾 陽一, 細川 勲, VPcogma: 組み込み機器を対象としたバイナリポータブルなユビキタス環境構築ミドルウェア, 情報処理研究報. 2003-UBI-2, No.115, pp.97-98 (2003).
- [3] Nobuo Kawaguchi, VPcogma: A Light-Weight Cooperative Middleware for Ubiquitous Embedded Devices, International Workshop on Software Architectures for Self-Organization (SASO'05) with Pervasive2005, (2005).
- [4] cogma プロジェクト <http://www.cogma.org/>
- [5] NEXCESS <http://www.nexcess.itc.nagoya-u.ac.jp/>