

LinkCodget: Web サーバ上の移動ソフトウェアを用いた機器間連携機構

春原 雅志[†] 河川 信夫[‡]

[†]名古屋大学大学院情報科学研究科 [‡]名古屋大学情報連携基盤センター

LinkCodget: Simple Device Coordination using Mobile Software on Web Server

Masashi Sunohara[†] Nobuo Kawaguchi[‡]

[†]Graduate School of Information Science, Nagoya University

[‡]Information Technology Center, Nagoya University

1. はじめに

近年、Web サーバが内蔵されたネットワーク対応の組み込み機器が増え、Web ブラウザを用いて設定や操作を行うことが一般的になりつつある。ユーザは専用のソフトウェアを用意する必要がなく、普段利用している Web ブラウザから機器の操作が可能のため、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 サーバの機能が内蔵されると、機器を操作するたびに、それぞれの Web サーバにアクセスしなければならないなどの煩雑さが生じる。また、温度センサのような比較的単純な組み込み機器に対して、Web サーバの機能を実装することは現実的ではない。さらに、個々の Web サーバに異なる IP アドレスを設定しなければならないという問題も生じる。このような課題に対し、我々は、移動ソフトウェア技術を用い、一台の Web サーバを用いて機器の集中管理を行う手法として WebCodget システム[1]を提案している。

WebCodget システムでは、各組み込み機器に搭載した移動ソフトウェアが Web サーバへ動的に移動し、ユーザに機器の情報や操作するためのインタフェースを提供する。移動ソフトウェアがサーバに移動するための基盤ソフトウェアとしては cogma[2]を用いている。ユーザは Web ブラウザを用いて Web サーバへアクセスすることで、機器を操作することが可能である。WebCodget システムにより、単一の Web サーバ上で複数の機器の操作が可能になるが、一連の操作を行う際には、個々のインタフェースを個別に操作し、同様の操作を繰り返す煩雑さがある。例えば、プレゼンテーションを行う場面を考えると、(1)電動スクリーンを下げる、(2)プロジェクタの電源を入れる、(3)部屋の照明を落とす、といった操作が必要である。このように日常生活においては複数機器に対する一連の操作を繰り返し行う事が少なくない。このような繰り返し頻度が高い操作は、ユーザが事前に操作を登録しておくことにより、次回以降、同様の操作をする際に再利用できると便利である。

本論文では、WebCodget システム対応の複数の機器に対して、繰り返し操作を支援する手法を提案する。本手法では、ユーザが WebCodget システムを用いて機器の操作を行う際、Web ブラウザから制御情報を含んだ POST データを Web サーバへ送る点に着目した。ユーザが機器を操作した際に、Web サーバを経由した POST データを記録・再現することにより、複数の機器に対する一連の繰り返し操作を支援する。

本論文の構成は以下の通りである。まず第 2 章で WebCodget システムについて説明し、第 3 章で WebCodget システム上にて機器連携を行いユーザの操作支援を行う手法について述べる。第 4 章で本論文についてまとめ、今後の課題を挙げる

2. WebCodget システム

2.1. WebCodget システムの概要

WebCodget システムは cogma[2]上のアプリケーションとして実装されている。cogma は我々の研究室で開発を進めてきた動的な環境変化に対応可能な基盤ソフトウェアである。cogma を用いることによりネットワーク上でのソフトウェアの移動が容易に行える。cogma についての詳細は 2.2 節で述べる。

WebCodget システムでは、各機器に WebCodget と呼ばれる移動ソフトウェアを搭載する。移動ソフトウェアはネットワークを通じて自身が移動し、機器の制御や情報のやりとりを行う。WebCodget には機器を制御するための機器固有の情報と Web ページを生成するための HTML データを含んでいる。WebCodget はネットワーク上の Web サーバ機能を持った WebServerCodget へ移動する。機器を操作するユーザが WebServerCodget に Web ブラウザからアクセスすることにより、機器の情報が表示され、Web ブラウザから機器の操作や設定が可能となる。図 1 に WebCodget システムの概要を示す。例えば、ライトのような単純な機器に Web サーバ機能を実装し、Web ブラウザから操作する場合、個々のライトに直接アクセスする必要が生じる。一方 WebCodget システムを用いると、個々のライトには、ライトを制御する機能と HTML データを含む WebCodget を搭載する。WebCodget は自動的に WebServerCodget を発見し移動する。ライトを制御するユーザは WebServerCodget にアクセスすることにより、複数のライトやその他の機器、センサ情報などを確認することが可能となる。

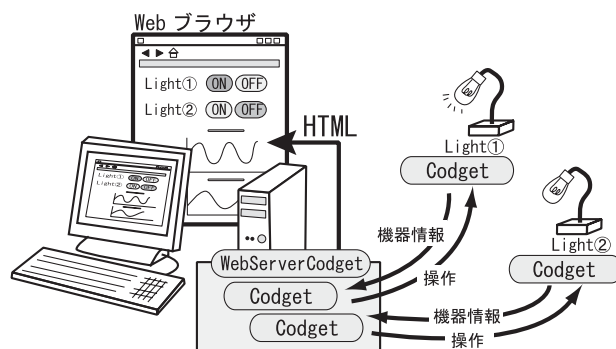


図 1 WebCodget システムの概念図

2.2. cogma: Mobile Software System

本節では WebCodget システムのベースとなっているユビキタスマドルウェア cogma (Cooperative Gadget for Mobile Appliances) について述べる。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) 複数の異種間ネットワークの同時利用が可能
 - TCP/IP、シリアル、HTTP 等の通信プロトコルを利用可能
- D) 他のノードの自律的な発見機構
 - Hello Packet 等の利用により、別ノードを自動発見
- E) 移動ソフトウェアの管理・通信機構
 - ノードの動的な追加・削除に対応
- F) 多様なプラットフォーム上での動作
 - VPcogma[3] [4]により様々な OS、CPU 上で動作

cogma では移動ソフトウェアの利用を基本としているため、ソフトウェアを必要に応じて動的にインストールすることが可能になる。WebCodget システムでは、操作対象の機器から Web サーバ上へ機能を移動させるために移動ソフトウェアを利用している。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、組込み機器向けのサービス発見機構を利用している。

2.3. WebCodget システムの構成

WebCodget システムは、主に以下の二つのソフトウェアから構成されている。一つは Web サーバの機能を備えた WebServerCodget であり、二つめは機器に内蔵する WebCodget である。図 2 に WebCodget システムの構成を示す。WebCodget を内蔵した機器がネットワークに接続されると、WebCodget は WebServerCodget へ移動する。ユーザが機器の状態を監視、制御する際には Web ブラウザを用いて WebServerCodget にアクセスし、Get や Post といったコマンドを用いて命令を送る。WebServerCodget がコマンドを受け取ると、サーバ上に存在する WebCodget に情報を渡す。サーバ上の WebCodget は機器側に移動し、機器の制御を行う。また、機器側で状態変化が発生すると、機器側の WebCodget がサーバに移動し、サーバが WebCodget から機器の情報を得ることにより Web ページの情報が更新される。

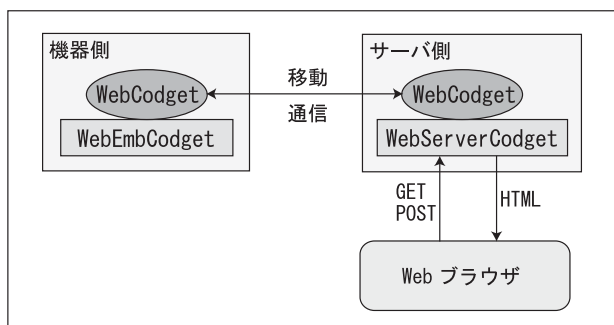


図 2 WebCodget システムの構成

2.3.1. WebServerCodget

WebServerCodget はネットワーク上の WebCodget をサーバ上に保持し、ユーザから Web ブラウザでアクセスされると HTML データを返す。WebServerCodget のベースとなっている cogma にはネットワーク上のサービスを自動発見する機構として EmbeddedManager という仕組みが存在する。EmbeddedManager は WebCodget が WebServerCodget を自動的に発見するために利用されている。それぞれの機器に搭載された WebCodget は EmbeddedManager を通じて、ネットワーク上の WebServerCodget を発見することが可能となる。

2.3.2. WebCodget

WebCodget は監視や制御を行う機器に搭載する移動ソフトウェアである。WebCodget が EmbeddedManager を通じて WebServerCodget を発見すると、自身のコピーを WebServerCodget へ転送する。ユーザが Web ブラウザで WebServerCodget へアクセスすると、WebCodget 内に格納されている HTML ページが表示される。また、ユーザが Web ブラウザから機器を制御するコマンドを送ると、WebServerCodget は WebCodget に対してコントロールメッセージを送り、メッセージを受け取った WebCodget が機器側へ移動することで機器の制御が行われる。このように、機器側のノードと Web サーバ側のノードとのやりとりは WebCodget がそれぞれのノード間を移動することによって実現される。

2.4. 実環境での WebCodget システム利用

WebCodget システムを実環境の機器に実装した例を図 3 に示す。図 3 は我々の研究室にある機器と、それらを制御するための操作インタフェースを示したものである。通常、ユーザがこれらの機器を操作する際には、機器ごとに実装されたスイッチや赤外線リモコンなどを利用する(表 1)。我々はこれらの機器をシリアル通信を介してネットワークに接続し、WebCodget システムの実装を行った。WebCodget を実装することにより、機器の状態監視や制御を Web ブラウザから行えるようになる。図 4 は Web ブラウザから WebServerCodget へアクセスした際に表示される機器操作の画面である。例えば、この Web ブラウザの画面上に表示されているプロジェクタコントローラのボタンを押すことにより操作が可能となる。

表 1 機器の操作インタフェースと操作内容

機器	操作インタフェース	操作内容
プロジェクタ	赤外線リモコン	電源 ON/OFF、入力切り替え等
ライト	押しボタンスイッチ	電源 ON/OFF
電動スクリーン	押しボタンスイッチ	スクリーン上下
電気スタンド	スライドスイッチ	電源 ON/OFF
RGB スイッチャ	押しボタンスイッチ	モニタ入力切り替え

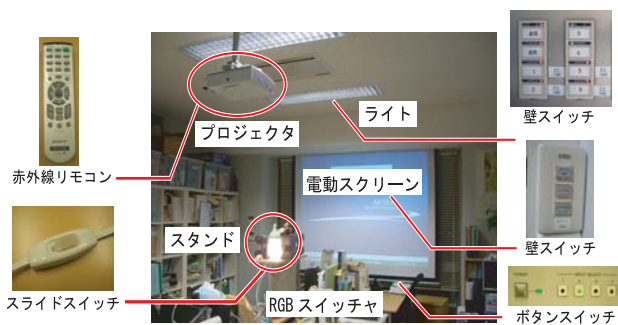


図 3 実環境にある機器

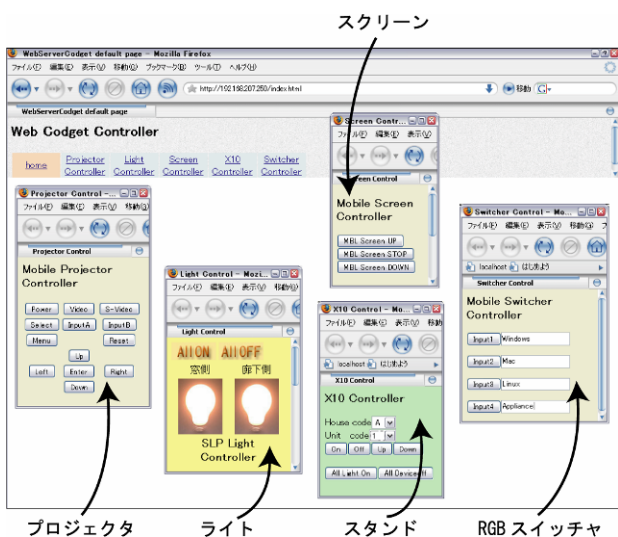


図 4 WebCodget システムによる操作画面

3. WebCodget システムでの機器間連携

3.1. WebCodget システムでの複数機器利用

WebCodget システムを用いることにより、ユーザは一台の Web サーバにアクセスすることにより、複数の機器の監視や操作が可能となる。しかし我々の生活において、単一の機器利用だけでなく、複数の機器を組み合わせる場面がたびたび見られる。このような場面では、ユーザは複数の機器に対して、毎回機器操作を行う必要がある。例えば、プロジェクタを用いてプレゼンテーションを行う場合、次のような操作が必要となる。

1. 電動スクリーンを下げる
2. プロジェクタの電源を入れる
3. スクリーンを見やすくするためにいくつかの照明を消す

これらの操作は、プレゼンテーションを行うたびに毎回それぞれの機器に対して個別に操作を行う必要がある。同様にプレゼンテーション終了時にもこれらの機器に対する操作が必要になる。そこでユーザの機器操作を記憶・再現することが可能な WebCodget を実装することにより、ユーザの機器操作支援が可能であると考えられる。次節より WebCodget システム対応の複数の機器に対して、繰り返し操作を支援する手法を述べ、その手法を実現するためのシステムを設計し、実装を行う。

3.2. 繰り返し操作支援の手法

WebCodget システムでは、Web ブラウザを通して機器操作を行う。ユーザはブラウザ上に表示された機器操作の画面のドロップダウンリストから操作項目を選択したり、操作ボタンをクリックしたりすることにより機器操作を行う。この時、機器に対する操作内容は POST データを用いて WebServerCodget へ送られる。POST データを受け取った WebServerCodget は、サーバ上に存在する操作対象の WebCodget に POST データを渡すことにより機器の操作が行われる。つまり、ユーザが機器操作を行う際には、Web ブラウザと WebServerCodget 間で POST データのやりとりが行われることになる。複数の機器に対して操作を行う際には、Web ブラウザから操作の指示を行うたびに POST データが WebServerCodget へ送られる。そこで、WebServerCodget へ送る POST データを記憶・再現可能なソフトウェアを設計し、複数の機器に対する操作を支援する手法を提案する。本研究では、ユーザが機器操作を行った際の POST データを記録し、再現可能なソフトウェアを LinkCodget と名付けた。ユーザは、単一の機器または複数の機器に対する一連の操作を、あらかじめ LinkCodget に登録しておき、必要に応じて LinkCodget から呼び出すことにより、個別の機器に対して同様の操作を毎回繰り返す必要が無くなると考えられる。

3.3. LinkCodget の設計と実装

3.2 節で述べた手法を元に LinkCodget の設計を行った。LinkCodget は主に以下の機能を持つ必要がある。

- (1) ユーザが WebCodget システム上の機器を操作した際の一連の POST データを記録する機能
- (2) 記録された操作を Web ブラウザを通してユーザに提示する機能
- (3) ユーザが記録した操作を再現する指示をした際に、WebServerCodget に対してそれぞれの機器を操作する POST データを送信する機能

これらの機能を持つ LinkCodget を Java を用いて実装した。LinkCodget は WebCodget システム上のアプリケーションとして実装したため、POST データを用いて操作を行う既存の WebCodget には変更を加える必要がない。また、WebServerCodget 上に存在させることにより、複数人で記録した操作を共有することが可能である。

3.4. LinkCodget への操作記録

複数の機器に対する一連の操作を LinkCodget へ記録する際の流れを図 5 に示す。LinkCodget への操作記録は以下の手順により行われる。

- (0) ユーザは WebServerCodget へ Web ブラウザを用いてアクセスし、LinkCodget の操作画面を表示させる。
- (1) ユーザは記録モードを開始するために、LinkCodget の「開始」ボタンを押す。
- (2) ユーザは Web ブラウザから操作を記録したい機器の WebCodget を操作する。
- (3) LinkCodget が記録モードになっている間は、POST データが LinkCodget へ送られる。
- (4) 操作一覧が LinkCodget へ送られ、Web ページ上に表示される。ユーザは必要なだけ(2)~(4)を繰り返す。

- (5) ユーザは記録したい操作を選択し、名前と共に一連の操作を定義する
- (6) 操作が LinkCodget へ記録され、記録モードは自動的に解除される。

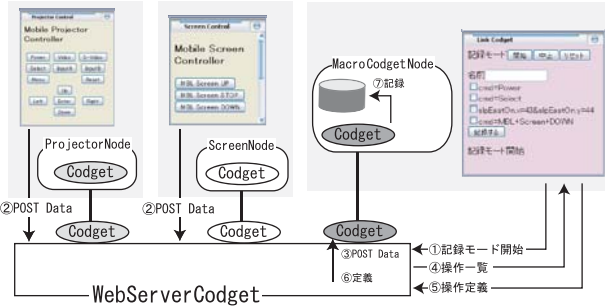


図 5 LinkCodget への操作記録

3.5. LinkCodget を用いた操作の再現

LinkCodget に記録した操作の再現は以下の手順により行われる。

- (0) ユーザは WebServerCodget へ Web ブラウザを用いてアクセスし、LinkCodget の操作画面を表示させる。
- (1) LinkCodget の操作画面に記録されている操作の一覧が表示される。
- (2) ユーザは再現したい操作を選択し、実行する。
- (3) LinkCodget から WebServerCodget に対して POST データが送られ、それぞれの機器が操作される。

3.6. LinkCodget を用いた実際の利用例

実際に LinkCodget を用いて操作を記録した例を図 6 に示す。図 6 では 3.1 節で取り上げた、プレゼンテーションを行う際の機器操作を LinkCodget へ登録した画面である。LinkCodget を記録モードにし、Web ブラウザを用いて電動スクリーン、プロジェクタ、部屋の照明の WebCodget に対して機器操作を行い「プレゼンテーション開始」という名前と共に定義した。次回以降は LinkCodget 上に表示されている「プレゼンテーション開始」ボタンを押すことにより、それぞれの機器操作が行われる。



図 6 LinkCodget への操作記録

4. まとめ

本論文では、WebCodget システム上で複数の機器を連携させて利用するためのアプリケーションとして LinkCodget を提案した。WebCodget システムは移動ソフトウェアを Web サーバに移動させることにより、複数の機器を単一の Web サーバで管理することが可能になった。それぞれの機器に Web サーバを実装する必要が無いため、機器が必要とするリソースは少なく済み、低コスト化が実現可能である。さらに、WebCodget システム上で LinkCodget を利用することにより、WebCodget システムで操作可能な機器に対する一連の動作を記憶・再現可能となった。LinkCodget は POST データを記録・再現する手法を用いているため、既存の WebCodget には新たな実装や変更を加えずに利用可能である。LinkCodget をサーバ上に保持することにより、複数人で登録した操作の共有が可能である。また、利用者側の端末で LinkCodget を保持することで、ユーザごとに操作内容を記憶・再現するという利用も考えられる。

今後の課題としては、WebCodget 同士が直接連携する仕組みを考えることがあげられる。現在の LinkCodget で操作を再現するには、LinkCodget から WebServerCodget へ POST データを受け渡し、それぞれの機器が操作されている。しかしこの方法では、常に WebServerCodget が必要であり、ネットワーク上から WebServerCodget を取り除いてしまうと LinkCodget が利用できなくなってしまうという問題点がある。また、SSLab[5]や Aware Home[6]といったスマートルーム内において、多数のセンサや機器を連携させた機器利用についても検討していきたいと考えている。

参考文献

- [1] 河川信夫, 春原雅志, 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, 情報処理学会ユビキタスコンピューティング研究会, 2005-UBI-009(2005).
- [2] 河川信夫, 稲垣康善: cogma: 動的ネットワーク環境における組み込み機器間の連携用ミドルウェア, 情報処理学会コンピュータシステム・シンポジウム, pp.1-8 (2001).
- [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] 河川信夫, 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, 情報研報. 2003-UBI-2, No.115, pp.97-98, (2003).
- [5] Okoshi, T., Wakayama, S., Sugita, Y., Iwamoto, T., Nakazawa, J., Nagata, T., Furusaka, D., Iwai, M., Kusumoto, A., Harashima, N., Yura, J., Nishio, N., Tobe, Y., Ikeda, Y. and Tokuda, H.: Smart Space Laboratory Project: Toward the Next Generation Computing Environment, IEEE Third Workshop on Networked Appliances (IWNA), (2001).
- [6] Abowd, G. A. Bobick, I. Essa, E. Mynatt, and W. Rogers, bibl.: The Aware Home: Developing Technologies for Successful Aging, American Association of Artificial Intelligence (AAAI) Conference 2002, Alberta, Canada, July 2002.