

視覚的な分散アプリケーション構築ツール uBlock の開発

uBlocks: Visual Distributed Application Builder for ubiquitous Environment

岩井将行¹⁾ 由良淳一²⁾ 望月祐洋³⁾
Masayuki Iwai Yura Junichi Yasuhiro Mochizuki

- 1) 慶應義塾大学 政策・メディア研究科 (〒252-8520 神奈川県藤沢市遠藤 5322 tailor@sfc.keio.ac.jp)
- 2) 慶應義塾大学 政策・メディア研究科 (〒252-8520 神奈川県藤沢市遠藤 5322 yurayura@sfc.keio.ac.jp)
- 3) 東京工業大学 学術国際情報センター (〒152-8550 東京都目黒区大岡山 2-12-1 moma@gsic.titech.ac.jp)

ABSTRACT. In an ubiquitous computing environment, both management and creation cost of distributed applications among numerous software components are becoming an important factor. A system which enables users to create applications on their own is especially needed, in a home area ubiquitous environment, to avoid loss of time and money. In this paper, we developed a software named uBlocks which can create distributed applications flexibly through multiple and multimodal user interface: Java 2D/3D-based, WEB-based, and character-based interface. As uBlocks is implemented based on Independent Modeling Topology, each interface is able to run simultaneously. This feature has much value to users in home ubiquitous environments.

1. 背景

近年、ネットワークインタフェースを持つ情報家電機器やセンサの小型化により、数多くのセンサや情報家電機器を室内の壁や家具に埋め込んだユビキタス環境が実現可能になりつつある。しかし、家電メーカ等が、このようなユビキタス環境をターゲットとした製品を製造する時、他の目的や仕様の製品との連携を想定することは開発コストを押し上げてしまい、非常に重荷となる。また、利用者はこのようなユビキタス空間において、ますます個々のソフトウェアの管理が煩雑になり、利用者自身がユビキタス空間全体を把握しておくことが困難となる。

我々は上記の課題を解決するために、ユーザによる分散アプリケーションの構築を視覚的に支援するソフトウェア uBlocks^{*1}(Ubiquitous Application Building Blocks Tools)の研究開発を行った [11][10][12]。uBlocks の目的は、一般的なユーザが分散するサービスコンポーネントを容易に組み合わせ、楽しみながら分散アプリケーションを構築可能にすることである。Java Beans[6] や Visual Studio などの GUI ベースのアプリケーション開発ツールとは異なり、uBlocks は完全に分散されているサービスコンポーネントを一つのアイコンとして扱う。実行中のアプリケーションもアイコン同士をつなぎかえるという簡略な操作によって、新たな分散アプリケーションを構成・再構成することが可能である。uBlocks は、ブロックを組み立てる様に、ユーザが自由に分散されるサービスを組み合わせ、分散アプリケーション構築の創造性が楽しめるソフトウェアである。

図 1 に、ユーザが実際ユビキタス空間において分散アプリケーションを視覚的なブラウザで構築している様子を示す。このようにユーザがタッチパネル等を利用して様々な情報家電機器やセンサのアイコンを簡単なマウス操作により結びあわせて、ユビキタス環境に適したアプリケーションを構築できる。この図においてユーザは、「無線 I D タ

グと室内のライト、エアコン、TV を接続」している。タグを保持したユーザを玄関先で検知し、ライト、エアコン、TV を連動させることが可能になる。切り離しも同様にマウス操作で行う。これによりメーカや目的の異なる情報家電機器間を接続でき、分散アプリケーションを構築できる。

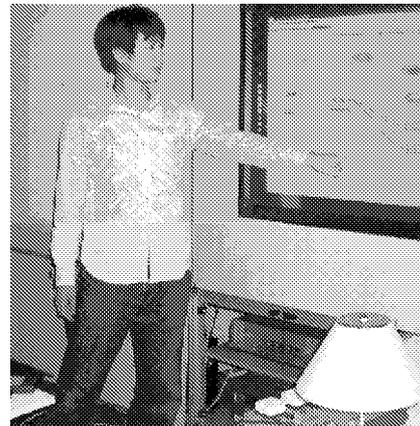


図 1: タッチパネルにより構成したアイコンの組み合わせがすぐさま実世界の分散アプリケーションに反映される。

本稿で述べる uBlocks は、複数種類のユーザからの操作インタフェースを持つ。Web ブラウザの Flash や携帯端末からの操作を可能にすることで uBlocks の利便性はさらに広がり、ユビキタス環境の利用方法が広げられる。また、センサや情報家電などを生産するハードウェアメーカは、uBlocks の API をもとに最低限のコストでユビキタス環境対応の製品を開発できる。

^{*1} 本ソフトウェア開発は平成 14 年度未踏ソフトウェア創造事業にて 2002 年 7 月から 2003 年 2 月の期間行われた。(萩谷 PMI 8-16, 14 情経第 962 号)

2. uBlocks の目的

2.1 現状課題の整理

近年、ネットワークインタフェースを持つ情報家電機器やセンサの小型化が著しい。例えば Dallas Semiconductor 社の TINI[1] は重量 180 グラムしかないが、Java プログラムの実行環境を提供する。こういった小型のプログラム実行環境は、今後さらに室内の壁の内部や白物家電に組み込まれていくと予想できる。

しかし、こういったユビキタス環境を想定した製品の投入を家電メーカー等が考える際に、いくつかの問題が発生する。現状のソフトウェア技術では、情報家電を製造する段階でネットワークを介して協調作業を行う相手を決定しなければならず、汎用性を失う。ユーザが季節や曜日ごとに動作させたいユビキタス空間における分散アプリケーションは異なるため、ユーザを考えた情報家電作りは複雑性を増すばかりで、開発者のコストを押し上げる。

さらに、ユーザ側は高度な設定を要求されるシステムを使うことになるため、負荷がかかってしまう。このため、ユビキタス環境が実現可能になったとしても、ユーザはすでに開発者が構築した、環境に適合でない分散アプリケーションを利用することを強いられる。

以上のように、ユビキタス環境では、情報家電メーカーがプログラム開発費のコスト増加の問題とユーザへの複雑なマニュアルを押し付けなければならないジレンマが生じている。

2.2 目的

我々は、上記の課題を解決するために、ユーザによる分散アプリケーションの構築を視覚的に支援するソフトウェア uBlocks (Ubiquitous Application Building Blocks Tools) の研究開発を行った。uBlocks の目的は、一般的な知識のないユーザが、容易に分散サービスコンポーネントを組み合わせ、楽しみながら、自らが分散アプリケーションを構築可能にすることである。

uBlocks は、ブロックのおもちゃで遊ぶ様な気軽さで、分散されたアプリケーションの構築を可能にしたソフトウェアである。ユーザ不在のユビキタス空間に関する研究が多い中、我々のソフトウェアは、ターゲットをユーザ主導のユビキタス環境構築に置いている。さらに、センサや情報家電などを生産するハードウェアメーカーが最低限のコストでユビキタス環境対応の製品を開発できる API を uBlocks は提供する。このように、uBlocks は情報家電機器業界の活性化につながるソフトウェアといえる。

2.3 関連研究および製品との比較

Java Beans(Sun Microsystems) や Visual Studio(Microsoft) などの GUI ベースのアプリケーション開発ツールとは異なり、uBlocks は、実行中の分散アプリケーションを簡略な操作で即興的に組み替えることが可能である。

また、AV 情報家電機器に特化した HAVi Event Service(HAVi group)[5], UPhP(Microsoft)[9], Carp@[2] などのミドルウェアがあるが、いずれも複数の情報家電機器(サービス)間での協調について考慮されていない。このため、一つのサービスに関する構成の柔軟性に乏しく、多様な協調作業を実現できない。我々の uBlocks は、複数の仲介者がイベントを中継し、目的の異なったソフトウェア同士を接続するための適応性を持った分散アプリケーションを構築できる。さらに uBlocks は、携帯端末やブラウザ、Java などの豊富なインタフェースを提供する。

3. 開発内容

uBlocks は、uBlock-Browsers, Dragon, TranService の 3 つのソフトウェア群から構成される。実装では Java および Jini[7] を用いる。uBlock-Browsers は、多種類のインタフェースに対応できるようにモデル生成の部分とインタフェースを表示する部分が分離しており、プロトコル非

依存になっているのが特徴である。また Dragon は、ユーザが分散アプリケーションを構築する際のサービス間通信手法の複雑性を吸収する。さらに、ユーザが接続できないサービス間を設定した場合に TranService が動的にその間を中継する変換サービスを探し出し、ユーザに知れることなく通信を成立させる。図 2 に uBlocks を構成するソフトウェア群の関係を示す。

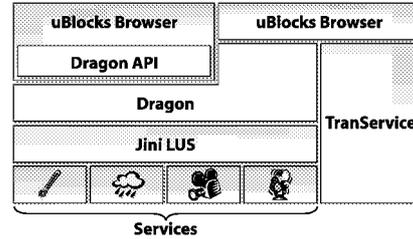


図 2: uBlocks 構成図

本節では、各ソフトウェア群の設計および実装について詳細を述べる。

3.1 uBlock-Browsers の設計・実装

uBlock-Browsers は、ユーザが実際にサービスを接続する際に利用するブラウザ群である。このブラウザ群は、Java/Swing, 実世界指向な 3D-GUI, i-mode などの携帯端末用など、さまざまなユーザインタフェースによって実装されている。図 3 に、Java/Swing による uBlock-Browser を示す。

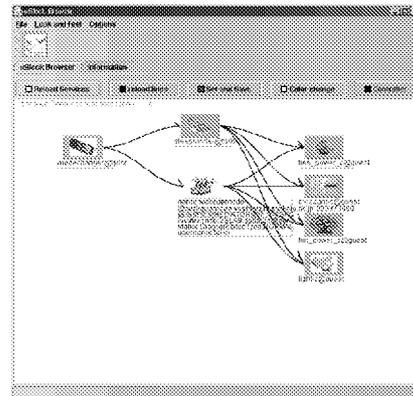


図 3: Java/Swing による uBlock-Browser

マルチモーダルインタフェースをサポートすることにより、uBlocks は、いつでも、どこでも操作可能になる。特に携帯端末向けインタフェースは、Java の実行環境がない環境においても表現力豊かなインタフェースを提供できる。マルチモーダルインタフェース機能を実現するために我々は、Web サービスに組み込む形で、uBlock-Browser を開発した。図 4 に示すように、uBlock-Browser は、HTTP などによって遠隔から接続されることを要求するインタフェースに現時点での操作可能なサービス群をユビキタス環境から集め表示する。この際、UI-Base は、分散サービス間の接続を行列式モデルによって保持する。ユーザからの分散アプリケーションを設定後 UI-Base は、各サービスが理解できるコマンド命令変化し各サービスに接続命令を発行する。接続命令は、次節に示す分散ミドルウェア Dragon の API を利用できる。

図 5 に、携帯端末用 uBlock-Browser の操作方法を示す。携帯電話により、uBlock サーバに接続を行い、接続元サービス一覧を表示させる (1)。その中から、接続したいサービスを選択する (2) と、そのサービスから接続可能なサービスの一覧が表示される。一覧には、現在接続され

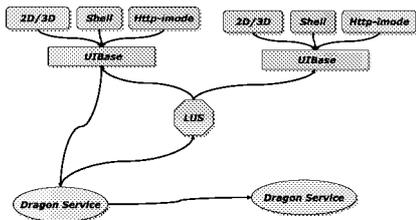


図 4: マルチモーダルな uBlock-Browsers との通信

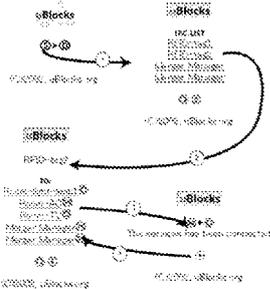


図 5: 携帯電話での画面遷移

ているか切断されているかも表示されている。あるサービスについて切断を選択する (3) と、そのサービスとの接続が切断される。これらの操作は、図 6 に示すように、Java/Swing の uBlock-Browser と連携できる。

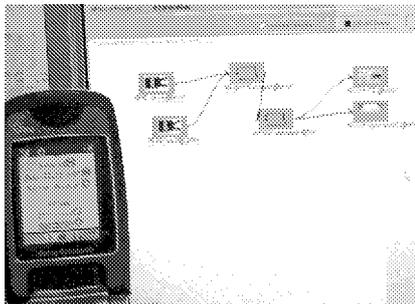


図 6: 携帯電話からの uBlock-Browser の操作

3.2 Dragon の設計・実装

uBlock-Browsers から発行された接続命令は、分散イベント配送ミドルウェア Dragon に通知される。分散されるサービス間の制御情報 (イベント) は、Dragon を通し Java オブジェクトのイベント配送によって協調動作を行う。多様なサービスが偏在するユビキタス空間では、単純に制御情報は、一方通行ではなく、配送形態も多様である。分散イベント配送ミドルウェア Dragon は、push 型配送、実時間性がありかつ信頼性のある put 型配送、pull 型配送、callback 型配送、callbackpull 型配送などの豊富なサービス間通信手法を実現した。

- push(A,B,evt) A から B に通知を非同期で通知を行う
- pushR(A,B,evt) A から B に信頼性をもって非同期で通知を行う
- pull(A,B) B から A に対してイベントを同期的に要求する
- callBack(A,B) A が B に対して自らの参照を送付, B が A に対してイベントを送付する
- callBackPull(A,B) A が B に対して自らの参照を送付, B が A に対してイベントを要求する

これらの配送モデル定義はユーザから隠蔽され、Dragon ミドルウェア内部ですべて吸収される。Dragon で定義する API により開発者は、ユビキタスサービスを低コストで実装できる。

3.3 TranService の設計・実装

TranService は、サービス接続の際のデータ変換を自動的にを行い、接続不可能だったサービスを接続可能にする。

異種サービスがユーザによって接続されようとする時 uBlock-Browser は、TranService を呼び出す。図 7 に示すように、TranService は二つのサービス間のプロファイル XML ファイルを比較し、そのプロファイルの差異を吸収する変換サービスを選択し、そのサービスを活性化 (インスタンス化) させ、中継接続させる。二つ以上の変換サービスを直列につなぎ合わせることも可能である。

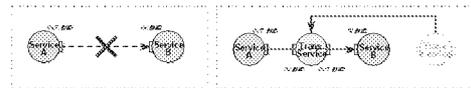


図 7: 携帯電話からの uBlock-Browser の操作

以下に PDA サービスプロファイルの例を示す。

```
<?xml version="1.0"?>
<qcs>
  <data>
    <type value="image">
      <subtype value="tbmp">
        <quality name="color">1</quality>
        <quality name="depth">0x2</quality>
        <quality name="width">180</quality>
        <quality name="height">180</quality>
      </subtype>
    </type>
  </data>
</qcs>
```

```
<type value="text">
  <subtype value="default">
    <quality name="charset">
      shift-jis</quality>
    </subtype>
  </type>
</data>
<protocol>
  <type value="http"></type>
</protocol>
</qcs>
```

このプロファイルを持つ PDA と動画を RTP[4] により配信できるカメラサービスをユーザが組み合わせた場合は、通常であれば、接続エラーを表示する。しかし TranService は、この異種の通信を補完する変換サービスを活性化し仲介させることが可能である。TranService は、対応しきれないほど数多くの多様な形式を意識しなければならなかった開発者のコストを軽減、変換サービスを流用することができる。

4. uBlocks ソフトウェア概要

uBlocks は、情報家電制御だけでなく、多様なソフトウェアコンポーネントによる分散アプリケーションを構築可能にするミドルウェアである。特別なハードウェア必要としないものから、工場等の制御に関わる特殊な機器制御システムまで幅広い分散システムが、このミドルウェアを用いて構成可能である。本節では、本ソフトウェアの配布パッケージの構成と各アプリケーションの詳しい説明を記述する。

5. uBlocks ソフトウェア概要

uBlocks は、情報家電制御だけでなく、多様なソフトウェアコンポーネントによる分散アプリケーションを構築可能にするミドルウェアである。特別なハードウェア必要としないものから、工場等の制御に関わる特殊な機器制御システムまで幅広い分散システムが、このミドルウェアを用いて構成可能である。本節では、本ソフトウェアの配布パッケージの構成と各アプリケーションの詳しい説明を記述する。

5.1 パッケージ構成

本ソフトウェアのパッケージ構成とその説明を以下に示す。

- **DRAGON:** Dragon およびアプリケーションのバイナリ, ソースコード
 - **bin:** Dragon/uBlocks Browsers の起動ファイル
 - **commands:** Dragon の基本的な設定ファイル
 - **htdocs:** WWW サーバのコンテンツ
 - **src**
 - **doc:** ドキュメント API
 - **drg**
 - **core:** Dragon コア部分
 - **event:** Dragon イベント配送部分
 - **service:** Dragon サービス部分
 - **view:** uBlocks Browsers 部分
 - **editor:** Java/Swing Browser 部分
 - **imode:** i-mode Browser 部分
 - **postit:** Java3D Browser 部分
 - **shell:** shell Browser 部分
 - **resources:** 画像など
 - **org**
 - **ublocks**
 - **app:** アプリケーション群
 - **filetransmitter:** ミーティングにおける資料配布支援
 - **modechanger:** リモコン一括機器制御システム
 - **ubiqinterphone:** 遠隔監視画像配信システム
 - **util:** アプリケーション共通のユーティリティ
- **TRANSERVICE:** TranService のバイナリ, ソースコード
 - **bin:** TranService 実行用スクリプト
 - **etc:** 変換モジュール群
 - **lib:** TranService ライブラリ
 - **src:** TranService のソースコード
 - **jp:** TranService 本体
 - **modules:** 変換モジュール

5.2 アプリケーション事例

本ソフトウェアの普及という点から、高価なハードウェアをなるべく必要としない3つのアプリケーションを提案、実装した。「ミーティングにおける資料配布支援」、「リモコン一括機器制御システム」、「遠隔監視画像配信システム」のそれぞれのアプリケーションは uBlocks 配布のパッケージに含まれており、ユーザが選択して利用できる。以下に各アプリケーションの詳細を述べる。

5.2.1 ミーティングにおける資料配布支援

必要機材

PC2 台以上、配布用のデジタルデータ (PPT ファイル, PDF ファイルなど)

目的と概要

ミーティングやプレゼンテーションなどではノート PC 持ち込みによるデジタルデータ用いたものが多くなっている。また、プレインストロミングなどでは、その場で2つの分散する資料ファイルを合成したり、必要なスライドを共有し新しい資料を作成していくといったことも行われる。本アプリケーションでは、分散するファイルを選択的に特定の相手に送りつけたり、特定の発表者のスライドを「今すぐ」もらってくるといったことがグラフィカルなブラウザ (uBlock-Browsers) を通して行える。さらに全員に資料を送りつけたり、ファイルを自分の自宅の PC などに

移動するといったことも可能である。また資料だけでなく URL を送りつけ、その URL を相手に開いてもらうなどといった、よりインタラクティブなミーティングが行える。実行手順

- 1) 各 PC で uBlock を起動し、Dropet サービスを読み込むと、uBlock-Browser 上に転送先 PC として表示される。
- 2) 画面にファイル転送用のアイコンが表示されるので、送付したいファイルをそのアイコンにドラッグする (図 8)。
- 3) uBlock-Browser 上で、ファイルと転送先 PC を接続する (図 9) と、転送先の PC 上に該当ファイルが表示される。

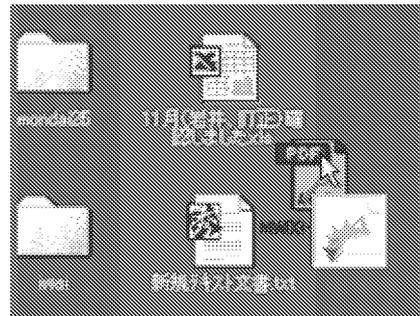


図 8: Dropet にファイルをドロップする

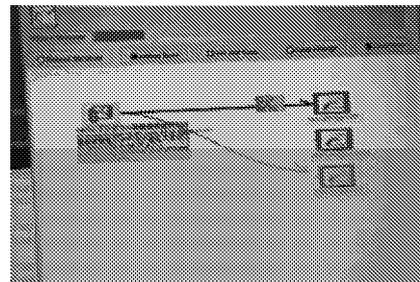


図 9: ファイルの転送設定を行う

5.2.2 リモコン一括機器制御システム

必要機材

PC1 台以上、制御要リモコン 1 個以上 (ハルコーポレーション製を推奨)

目的と概要

家庭内のリビング等にはリモコンで操作できる家電機器が数多くある。しかし、数多くのリモコンをそれぞれ使いこなさなければならないことはユーザにとって大きな負担である。本アプリケーションでは、まとまった処理を「モード」として表現し、それを選択するだけで家電の一括制御を行う。これにより、複数のリモコンを使って、複雑な操作をすることがなくなる。たとえば、「テレビとビデオとアンプの電源を同時に入れる/同時に切る」といった作業や、「エアコンを切って扇風機をつける/扇風機を入れてエアコンを切る」などの一連の動作のリモコン操作を一括で行うようにユーザは設定できる。

実行手順

- 1) 各 PC で uBlock を起動し、さまざまな機器サービスや、モード選択サービスを読み込むと、uBlock-Browser 上に各サービスが表示される。
- 2) uBlock-Browser 上で、モード選択サービスから各機器サービスに接続を行うことで、一括制御設定を行う (図 10)。

- 3) モード選択を行うことで、一括して機器制御を行うことができる。

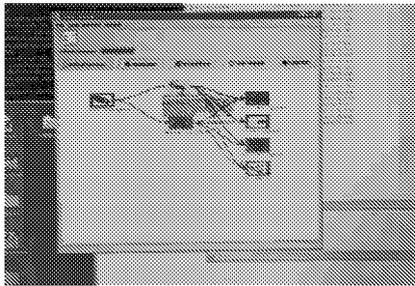


図 10: 一括制御を設定する

5.2.3 遠隔監視画像配信システム

必要機材

PC1台以上, USB 小型カメラ, i-mode 対応携帯電話(無くても良い)

目的と概要

本アプリケーションは、家庭にあるインタフォンに写る映像を、携帯電話に転送することを可能にする。たとえば外出先で、家庭への訪問者のことが気になったとする。ユーザは携帯電話の i-mode からその画像をこの携帯電話に転送することを設定する。さらに会社については会社の PC に映像を転送するように設定できる。画像は携帯電話や会社のブラウザにあうように適応的に変換を行って、画質を動的に下げる。

実行手順

- 1) 各 PC で uBlocks を起動し、カメラキャプチャサービス、転送先の PC または携帯電話サービスを読み込む。
- 2) uBlock-Browser 上で、キャプチャ画像を表示する方に接続を行う。
- 3) ボタンを押すことで、キャプチャされた画像が設定した通知先(PC や携帯電話) に表示される(図 11)。その際、携帯電話に通知した場合は、画面に適するようにサイズや色数が変更される(図 12)。



図 11: スクリーンへの通知



図 12: 携帯電話への通知

6. まとめ

uBlocks は、情報家電機器やセンサを自由に組み合わせてユーザが分散アプリケーションを視覚的に構築するためのミドルウェア及びインタフェースを提供する。これにより、プログラムの知識のない一般的なユーザが、ユビキタス環境をより容易にマウス操作で確認しながら構築できる。また本ソフトウェアの有効性を示すために、高価なハードウェアをなるべく必要としない3つのアプリケーション:「ミーティングにおける資料配布支援」、「リモコン一括機器制御システム」、「遠隔監視画像配信システム」を実装し運用した。

今後は、SOAP[3], EJB[8] などとの連携を図り、より応用範囲の広いミドルウェアを目指す。

7. 参加企業及び機関

なし

参考文献

- [1] Dallas Semiconductor Corp. Tiny InterNet Interface. <http://www.ibutton.com/TINI/index.html>.
- [2] Michael Fahrmaier, Chris Salzman, and Maurice Schoenmakers. A Reflection Based Tool for Observing Jini Services. In *Reflection and Software Engineering*, pp. 209-227, June 2000.
- [3] W3C Note. Simple Object Access Protocol (SOAP) 1.1, May 2000. <http://www.w3.org/TR/SOAP/>.
- [4] H. Schulzrinne, A. Rao, and R. Lanphier. Real Time Streaming Protocol (RTSP), 1998. RFC 2326.
- [5] Sony, Matsushita, Philips, Thomson, Hitachi, Toshiba, Sharp, and Grundig. Specification of the Home Audio/Video Interoperability (HAVi) Architecture, May 1998. <http://www.havi.org/home.html>.
- [6] Sun Microsystems Inc. JavaBeans specification, 1997. <http://www.java.sun.com/products/embeddedjava/>.
- [7] Sun Microsystems Inc. Jini Architecture Specification, October 2000. <http://www.sun.com/jini/specs/jini1.1.pdf>.
- [8] Sun Microsystems, Inc. Enterprise Java Beans Technology, 2001. <http://java.sun.com/products/ejb/>.
- [9] Universal Plug and Play Forum. Universal Plug and Play (UPnP), 1999. <http://www.upnp.org>.
- [10] 岩井将行, 中澤仁, 徳田英幸. ublocks: Enabling user-side composition of distributed ubiquitous application. 情報処理学会 情報家電コンピューティング研究会 (IAC-4), Vol. 4, pp. 29-36, November 2002.
- [11] 岩井将行, 中澤仁, 徳田英幸. 複数個インタフェースからの一貫性のある分散アプリケーション構築に関する研究. 日本ソフトウェア科学会ソフトウェアシステム研究会 SPA サマーワークショップ, October 2002.
- [12] 岩井将行, 中澤仁, 徳田英幸. 分散アプリケーション構築を複数種類インタフェースから実現可能にする研究. 日本ソフトウェア科学会 ソフトウェアシステム研究会 SPA, 2003.