

リモートディスプレイ環境における複数ユーザの 同時利用のためのアクセス制御機構

上 向 俊 晃[†] 原 隆 浩[†] 塚 本 昌 彦[†]
萩 野 浩 明^{††} 西 尾 章 治 郎[†]

本論文では、筆者らがこれまでに提案しているリモートディスプレイ環境において、複数のユーザが1つのリモートディスプレイに同時にアクセスして協調作業や対戦型ゲームを行うためのアクセス制御機構を提案する。提案機構では、共通ID、識別ID、および、トークンを使用してアクセス制御を行う。共通IDにより、リモートディスプレイを共有するユーザが利用しているアプリケーションを識別し、識別IDにより、各アプリケーションを利用しているユーザを識別する。さらにトークンにより、共通IDと識別IDを持つユーザに対してリモートディスプレイを同時に操作する権限を与える。共通IDと識別IDを持つユーザ間でトークンをやりとりすることにより、複数のユーザがリモートディスプレイを共有できる。

An Access Control Mechanism for Concurrent Accesses by Multiple Users in Remote Display Environments

TOSHIAKI UEMUKAI,[†] TAKAHIRO HARA,[†] MASAHIKO TSUKAMOTO,[†]
HIROAKI HAGINO^{††} and SHOJIRO NISHIO[†]

In this paper, we propose an access control mechanism for concurrent accesses to a *remote display* by multiple users in *remote display environments*. In this mechanism, we use the *common ID*, *personal ID*, and *token* for access control. The common ID is used to identify an application in which users share a remote display, while the personal ID is used to identify each user with the same common ID. The token is used to allow some users with both ID to operate the remote display. Multiple users with both IDs can share a remote display for an application such as cooperative work and match game by exchanging tokens among them.

1. はじめに

近年、コンピュータの小型化、軽量化、および、ネットワーク技術の発展により、モバイルコンピューティング環境が急速に普及している^{2),9)~11)}。モバイルコンピューティング環境では、ユーザは携帯可能なコンピュータを用いて、いつでもどこでも、移動しながらネットワークにアクセスでき、情報の通信や処理を行うことができる。モバイルコンピューティング機器としては、特に携帯電話の普及が著しい。今日の携帯電話は単に電話のための端末ではなく、E-mailやWWWブラウジング、スケジュール管理などの機能

を備えており、手のひらサイズのコンピュータとして認められつつある。しかし、携帯電話は、画面が小さく、一般のコンピュータのように多数の文字や画像を含む情報を一度に表示できない。

一方、ユビキタスコンピューティング環境に対する注目も高まっている^{1),5),8),14)}。ユビキタスコンピューティング環境では、生活空間のいたるところにコンピュータが存在し、だれでも自由にそのコンピュータを利用できる。現在、ユビキタスコンピューティング機器として、街角に設置されたタッチパネルディスプレイなどがある。しかし、これらは、アプリケーションに特化したシステム構築が行われているため、統一的なインフラストラクチャを整備するには、多額の資金が必要になる。そのため、一般のアプリケーション開発者が、ユビキタスコンピューティング環境上でサービスを展開することは困難である。

そこで筆者らは、モバイルコンピューティング環

[†] 大阪大学大学院工学研究科情報システム工学専攻
Department of Information System Engineering, Graduate School of Engineering, Osaka University

^{††} 株式会社NTTドコモマルチメディア研究所
Multimedia Laboratories, NTT DoCoMo, Inc.

境とユビキタスコンピューティング環境を融合したコンピュータの新しい利用環境として、リモートディスプレイ環境を提案した¹³⁾。リモートディスプレイ環境では、生活空間のいたるところに、インターネットに接続された大画面表示可能なディスプレイが存在する。このディスプレイをリモートディスプレイと呼ぶ。ユーザは、携帯電話をリモートコントローラのように利用してリモートディスプレイの画面操作を行い、情報の掲示や閲覧を行う。今日、携帯電話は非常に普及しているため、ユーザは新たにインタフェースを購入する必要がない。さらに、インターネットや携帯電話の既存のインフラストラクチャを利用できるため、リモートディスプレイ環境の実現は容易である。

さらに筆者らは、リモートディスプレイ環境の応用例の1つとして、WWWブラウジングシステムを実装した。現在の携帯電話のWWWブラウジング機能では、表示画面のサイズが限られているため、携帯電話での表示を考慮して作成されたWebページしか表示できない。そこでこのシステムでは、リモートディスプレイ上には通常のWebページを表示し、携帯電話上にはリモートディスプレイの画面操作を行うための操作ページを表示する。操作ページは、リモートディスプレイ上に表示する通常のWebページから、リンクとフォームのみを抽出することにより作成される。また、このシステムでは、1つのリモートディスプレイに対して1人のユーザのみが操作できるアクセス制御機構を実現した。

リモートディスプレイ環境のその他の応用例として、複数のユーザが1つのリモートディスプレイに同時にアクセスして、協調作業や対戦型ゲームを行うものが考えられる。そこで本論文では、そのような応用に対するアクセス制御機構を提案し、筆者らが実装したWWWブラウジングシステム上の実装について述べる。提案機構では、アプリケーション開発者がリモートディスプレイ上で利用するアプリケーションを容易に記述できるように、ライブラリとしていくつかの関数を提供した。さらに、提案したアクセス制御機構を用いて、複数のユーザが協調してリモートディスプレイを利用するアプリケーションを実装した。

以下では、2章で筆者らが以前実装したWWWブラウジングシステムについて述べる。3章では、複数ユーザからのアクセスを考慮したアクセス制御機構について述べる。4章では、提案機構に基づいて実装したアプリケーションについて述べる。5章では、提案機構について考察を行う。最後に、6章で結論と今後の課題を述べる。

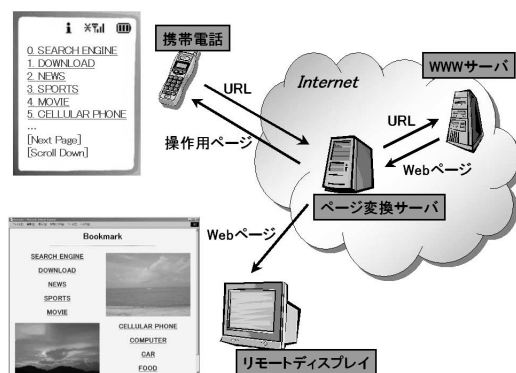


図1 WWWブラウジングシステムの構成図
Fig.1 Structure of the WWW browsing system.

2. リモートディスプレイ環境におけるWWWブラウジングシステム

筆者らが以前実装したWWWブラウジングシステム¹³⁾の構成を図1に示す。このシステムは、ユーザインタフェースとなる携帯電話、リモートディスプレイ、WWWサーバ、および、ユーザの入力に応じてリモートディスプレイと携帯電話のそれぞれにWebページを表示させるページ変換サーバの4つの要素からなる。携帯電話としては、端末単体でWWWブラウジング可能な機器を使用し、リモートディスプレイとしては、Webページを表示できるブラウザ機能を持つ通常のコンピュータディスプレイを使用する。さらに、WWWサーバは通常のものを使用する。したがって、ページ変換サーバのみを設計・実装した。

インターネットに接続された各リモートディスプレイには、一意な識別子としてURLが与えられており、ユーザはリモートディスプレイにアクセスする際、携帯電話からこのURLを指定する。このURLは、システムの初期ページに対応しており、その初期ページは対応するページ変換サーバを持つ。以降、携帯電話から送信される情報は、つねにそのページ変換サーバが処理を行う。ページ変換サーバは、WWWサーバにアクセスし、携帯電話から受信したURLに対応するWebページをリモートディスプレイ上に表示する。同時に、そのWebページからリンクとフォームだけを抽出することにより、そのWebページに対応する操作ページを自動作成し、携帯電話上に表示する。ユーザは、携帯電話に表示された操作ページを用いて、携帯電話では得られない大画面のディスプレイ上で通常のWebページを閲覧できる。

各リモートディスプレイは公共の場に設置されるた

め、1つのリモートディスプレイに対して複数のユーザが同時にアクセスすることが考えられる。そこで、このシステムでは、1人のユーザがあるリモートディスプレイにアクセスしている間は、そのユーザの操作を優先させ、他のユーザからのアクセスを拒否するアクセス制御機構を実現した。この機構では、ユーザにIDを割り当てることにより、リモートディスプレイを操作する権限を与える。IDを持つユーザが明示的にIDを解放するか、もしくは、そのユーザが一定時間リモートディスプレイにアクセスしない場合に、ページ変換サーバが強制的にIDを剥奪することにより、他のユーザによるリモートディスプレイの操作を可能とする。

3. 複数ユーザのためのアクセス制御機構

リモートディスプレイ環境のその他の応用例として、複数のユーザが1つのリモートディスプレイに同時にアクセスして、協調作業や対戦型ゲームを行うものが考えられる。本章では、そのような応用に対するアクセス制御機構を提案し、筆者らが実装したWWWブラウジングシステム上で行った実装について述べる。

3.1 アクセス制御機構の概要

提案機構では、リモートディスプレイを共有するユーザの認証を行うため、共通ID、識別ID、および、トークンを使用する。共通IDにより、リモートディスプレイを共有するユーザが利用しているアプリケーションを識別し、識別IDにより、各アプリケーションを利用しているユーザを識別する。さらにトークンにより、両IDを持つユーザに対してリモートディスプレイを同時に操作する権限を与える。このとき、トークンの有無によって、ユーザの携帯電話上に表示する操作用ページを変更する。トークンを持つユーザのための操作用ページは、リモートディスプレイを操作するためのリンクやフォームを含む。一方、トークンを持たないユーザのための操作用ページは、トークンを要求するためのリンクを含む。各アプリケーションの開発者は、共通IDの名称、リモートディスプレイを共有できるユーザの制限数として識別IDの最大数、および、同時にリモートディスプレイを操作できるユーザの制限数としてトークンの最大数を指定する。提案機構において、識別IDの最大数が2、トークンの最大数が1の場合の動作例を図2、図3に示す。図2は共通IDと識別IDを配布するときの動作を示し、図3はトークンを配布するときの動作を示している。

提案機構では、以下のようにして、複数ユーザのアクセス制御を行う。

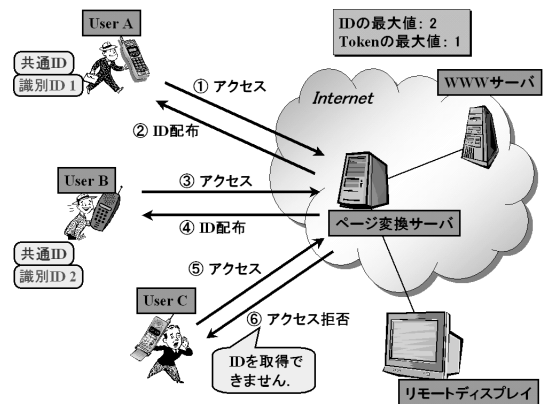


図2 アクセス制御機構(両IDの配布)

Fig. 2 The access control mechanism: distribution of both IDs.

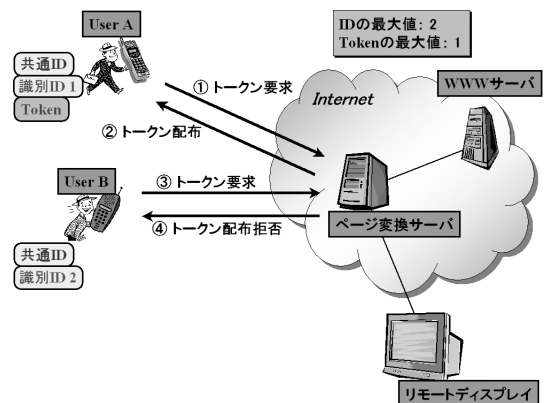


図3 アクセス制御機構(トークンの配布)

Fig. 3 The access control mechanism: distribution of token.

- (1) ユーザは、リモートディスプレイのURLを携帯電話上で入力し、携帯電話のWWWブラウジング機能を利用してそれにアクセスする。このURLは各アプリケーションの初期ページに対応する。以降、携帯電話に入力された情報はページ変換サーバに送られる。
- (2) ページ変換サーバはWWWサーバにアクセスし、携帯電話から受け取ったURLに対応するWebページ(初期ページ)を取得する。
- (3) アプリケーションを利用しているユーザ数が、そのアプリケーションで指定された識別IDの最大数に達していない場合、ページ変換サーバはユーザからのアクセスを許可し、そのユーザに共通IDと識別IDを与える。このとき、ページ変換サーバはそのユーザに対してトークンを与えず、携帯電話上にはトークンを要求するため

- のリンクを含んだ操作ページを表示する。もし、アプリケーションを利用しているユーザ数が、識別 ID の制限数に達している場合、ページ変換サーバはユーザからのアクセスを拒否する。
- (4) 共通 ID と識別 ID を持ち、トークンを持っていないユーザは、リモートディスプレイの操作を行いたいとき、操作ページ内のリンクを選択し、トークンを要求する。
- (5) ページ変換サーバは、トークンを要求してきたユーザの持つ共通 ID と識別 ID の認証を行う。すでにトークンを持っているユーザ数が、そのアプリケーションで指定されたトークンの最大数に達していない場合、ページ変換サーバはそのユーザにトークンを与える。さらに、ページ変換サーバは、そのユーザの携帯電話上にリモートディスプレイの操作を行うためのリンクやフォームを含む操作ページを表示する。
- (6) 両 ID とトークンを持つユーザは、携帯電話上でリンク選択やフォーム操作を行う。
- (7) ページ変換サーバは、ユーザの持つ両 ID とトークンの認証を行い、ユーザの操作に基づいてリモートディスプレイ上の Web ページを更新する。同時に、ページ変換サーバは、ユーザからトークンを剥奪し、携帯電話上にトークンを持たないユーザのための操作ページを表示する。(4)へ戻る。

このように、共通 ID と識別 ID を持つユーザは、トークンをやりとりすることにより、リモートディスプレイを共有できる。

3.2 アクセス制御機構の設計

本節では、提案機構の設計について説明する。筆者らがこれまでに実装したページ変換サーバは、ページ取得部、アクセス制御部、ページ変換部の 3 つのモジュールから構成される。以下では、提案機構を付加した各モジュールの設計について説明する。

[ページ取得部]

ユーザが携帯電話に入力した URL を受信し、WWW サーバから対応する Web ページの HTML ソースを取得する。さらに、その URL と取得したソースをアクセス制御部とページ変換部に渡す。

[アクセス制御部]

各アプリケーションで指定された識別 ID の最大数に達するまで、アクセスしてきたユーザに対して共通 ID と識別 ID を割り当てる。これらの ID は、両 ID を持つユーザが明示的に解放するとき、あるいは、両 ID を持つユーザがリモートディスプレイに一定時間

```
<HTML><HEAD>
<TITLE>Template of applications</TITLE>
<!--<RD comID="共通 ID"
      maxperID="識別 ID 最大数"
      token="トークン最大数">-->
<!--<RDLIB>ライブラリ関数</RDLIB>-->
</HEAD><BODY>
<!-- 通常の HTML ソース -->
</BODY></HTML>
```

図 4 アプリケーションの HTML テンプレート
Fig. 4 The template Web page for applications.

アクセスしないときに無効になる。さらにアクセス制御部は、各アプリケーションで指定されたトークン最大数に達するまで、トークンを要求してきたユーザに対してトークンを割り当てる。トークンを持つユーザがリモートディスプレイの操作を終えると、そのユーザからトークンを剥奪する。

アクセス制御部は、ユーザに割り当てた両 ID とトークンに関する情報をページ変換部に渡す。

[ページ変換部]

ページ取得部から受け取った HTML ソースと、アクセス制御部から受け取った両 ID とトークンに関する情報をもとに、操作ページを作成する。トークンを持つユーザのための操作ページは、HTML ソースからリンクとフォームだけを抽出して作成する。一方、トークンを持たないユーザのための操作ページは、トークンを要求するためのリンクを追加して作成する。このように作成した操作ページを携帯電話上に表示し、同時に、通常の Web ページをリモートディスプレイ上に表示する。

3.3 アクセス制御機構の実装

本節では、前節の設計に基づいたアクセス制御機構の実装について述べる。

提案機構を用いてアプリケーションを開発する際、図 4 に示すテンプレートに従って<HEAD>タグ内に必要事項を記述する。まず、3 行目のように、<RD>タグ内に共通 ID の名称、識別 ID の最大数、および、トークンの最大数を指定する。さらに、各アプリケーションの Web ページを容易に記述するために、ライブラリとしていくつかの関数を提供する。これらのライブラリ関数の呼び出しは、図 4 の 5 行目のように、<RDLIB>タグ内に記述する。提供するライブラリ関数は次のとおりである。

- TokenRequest

トークンを持たないユーザのための操作ページにトークンを要求するためのリンクを追加す

る．この関数は，すべてのアプリケーションで指定する．

- AddDocument(token, document)
ユーザがトークンを持つ場合 (token=1) と、トークンを持たない場合 (token=0) によって、操作ページに追加する文章や画像 (document) を指定する．さらに、トークンの有無によらず、ユーザの操作ページに共通の文章や画像を追加する場合は、token=2 とする．
- IDValidPeriod(period)
提案機構では、ユーザがリモートディスプレイに一定時間アクセスしない場合、そのユーザに割り当てた共通 ID と識別 ID を無効にする．この関数は、その有効時間を period に指定する．共通 ID が同じであれば、すべてのユーザの各 ID の有効時間は同じものとなる．指定のない場合は、デフォルト値として 60 秒になる．
- TokenValidPeriod(period)
トークンの有効時間を period に指定する．デフォルト値は 50 秒である．
- CompareVariable(var1, var2, document)
変数 var1 の値と変数 var2 の値を比較し、値が等しい場合、document を操作ページに追加する．
- TokenMethod(method)
両 ID を持つユーザに対するトークンの割当て方法を指定する．method=match の場合、両 ID を持つユーザ間でトークンを順番に巡回させる．また、method=random の場合、トークンを配布するユーザをランダムに決定する．さらに、method=first の場合、トークンを要求する順番が早いユーザから順番にトークンを割り当てる．
- AutoUpdate(url, period)
トークンを持つユーザが携帯電話上で行う操作に関係なく、リモートディスプレイ上の Web ページを period の時間間隔で自動更新させる．url は、そのときの初期ページの URL であり、“http://quiz.cgi?q=1” のように、引数には更新に必要な変数とその初期値を指定する．更新ごとにその値を 1 ずつ増加させる．

一方、ページ変換サーバの 3 つのモジュールは、以下のように実装した．

[ページ取得部]

要求された URL に対する Web ページの HTML ソースを取得するため、LWP ライブラリ¹²⁾の LWP::Simple クラスの “get” 関数を使用した．ページ取得部は、取得した HTML ソースをアクセス制御

部とページ変換部に渡す．

[アクセス制御部]

ページ取得部から受け取った HTML ソースから、<RD>タグ内に記述された共通 ID、識別 ID の最大数、トークンの最大数を抽出し、両 ID とトークンに関する情報をユーザに割り当てる．ユーザに割り当てられた両 ID とトークンに関する情報は、ページ変換部に渡され、操作ページ内のすべてのリンクの URL に引数として付加される．アクセス制御部は、ページ取得部から受け取った URL をもとにユーザの持つ両 ID とトークンの認証を行う．さらに、HTML ソース内の <RDLIB>タグ内に “IDValidPeriod(period)” 関数が記述されている場合は、両 ID の有効時間を period に設定し、“TokenValidPeriod(period)” 関数が記述されている場合は、トークンの有効時間を period に設定する．指定がない場合は、それぞれをデフォルト値に設定する．また、“TokenMethod(method)” 関数が記述されている場合、method に応じてトークンを要求するユーザに対してトークンを割り当てるかどうかを決定する．

[ページ変換部]

ページ取得部から受け取った HTML ソースから必要な部分を抽出し、2 種類の操作ページを作成する．トークンを持つユーザのための操作ページは、HTML ソースの <BODY>タグ内からリンクとフォームだけを抽出して作成する．このとき、<RDLIB>タグ内に “AddDocument(1, document1)” 関数が記述されている場合、ページ変換部は、その操作ページに document1 を追加する．一方、トークンを持たないユーザのための操作ページは、HTML ソースの <RDLIB>タグ内に “TokenRequest” 関数が記述されていることを確認し、トークンを要求するためのリンクを追加して作成する．このとき、<RDLIB>タグ内に “AddDocument(0, document2)” 関数が記述されている場合、ページ変換部は、その操作ページに document2 を追加する．さらに、HTML ソースの <RDLIB>タグ内に “AddDocument(2, document3)” 関数が記述されている場合、ユーザがトークンを持っているかどうかに関係なく、操作ページに document3 を追加する．

アクセス制御機構では、ユーザに割り当てた共通 ID、識別 ID、および、トークンに関する情報は、操作ページ内の各リンクの URL の引数として付加する．このとき、アプリケーションの Web ページに対応する URL に関する情報も、各リンクの URL の引数として付加する．図 5 に、操作ページ内のリンクの URL 例を示す．これは、共通 ID が “1” のユー

```
http://RDserver.cgi?url=http://othello.cgi
&comID=Othello&perID=1&token=1
```

図5 操作ページ内の URL 例

Fig. 5 An example of a URL in an operational page.

ザがトークンを持つ場合 (token=1) を表している . “http://othello.cgi” は , 共通 ID が “Othello” と定義されたアプリケーションの Web ページの URL を表し , “http://RDserver.cgi” は , ページ変換サーバ上の CGI プログラムの URL を表している .

4. アプリケーション例

提案機構に基づいて , 以下のような 3 つのアプリケーションを実装した .

4.1 オセロ

オセロは , 白と黒の 2 人のプレーヤによる戦略ゲームである . プレーヤは , 表裏白黒の駒を縦横 64 マスの盤上に置く . 一方のプレーヤが自分の色の駒を置き , その駒と自分の色の駒に挟まれた相手の駒は色が反転する . 最終的に , 駒の色が多い方が勝者となる . 図 6 にリモートディスプレイに表示される Web ページを示す . また , 図 7 にその HTML ソースを示す .

このアプリケーションを提案機構に基づいて実現するために , 共通 ID の名称を “Othello” , 識別 ID の最大数を “2” , トークンの最大数を “1” とした . また , 提案機構が提供する , “TokenRequest” , “AddDocument” , “IDValidPeriod” , “TokenValidPeriod” , “TokenMethod” 関数を使用した . “AddDocument” 関数により , トークンを持つユーザのための操作ページには “It is your turn.” を表示し , トークンを持たないユーザのための操作ページには “Wait a minute!” と表示した . “IDValidPeriod” 関数により , 共通 ID と識別 ID の有効時間を 60 秒とし , “TokenValidPeriod” 関数により , トークンの有効時間を 45 秒とした . したがって , トークンを持つユーザは 45 秒以内に駒を置かなければならない . “TokenMethod” 関数により , トークンを 2 人のユーザ間で交互に割り当てる . したがって , 一方が連続してリモートディスプレイを操作できないことになる .

トークンを持つユーザのための操作ページは , 図 8 のように , ユーザが置きたい駒の位置 (“vertical” と “horizontal”) を入力するためのフォームを含む . 一方 , トークンを持たないユーザのための操作ページは , 図 9 のように , トークンを要求するためのリンクを含む . 図 10 に前者の HTML ソースを , 図 11 に

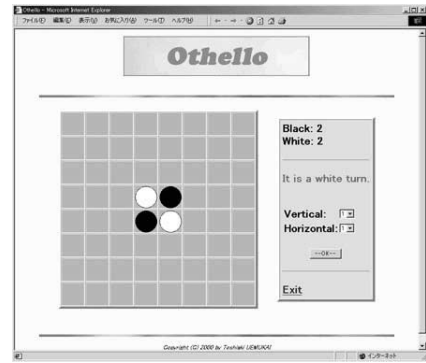


図6 アプリケーション例 : オセロ

Fig. 6 An application example: Othello.

```
<HTML><HEAD><TITLE>Othello</TITLE>
<!--<RD comID="Othello" maxperID="2"
token="1"-->
<!--<RDLIB>TokenRequest::
AddDocument(1,"It is your turn."):
AddDocument(0,"Wait a minute!"):
IDValidPeriod(60)::TokenValidPeriod(45)::
TokenMethod(match)</RDLIB-->
</HEAD><BODY>
.....
<FORM ACTION="http://othello.cgi">
Vertical: <SELECT NAME="vertical">
<OPTION VALUE="1">1 ..... </SELECT>
Horizontal: <SELECT NAME="horizontal">
<OPTION VALUE="1">1 ..... </SELECT>
<INPUT TYPE="hidden" NAME="color"
VALUE="white">
<INPUT TYPE="submit" VALUE="--OK--"></FORM>
.....
</BODY></HTML>
```

図7 オセロアプリケーションの HTML ソース (抜粋)

Fig. 7 A source code of the Othello application.

後者の HTML ソースを示す .

4.2 クイズ

雑学問題に答えるクイズである . 解答は 3 択になっており , ユーザは各問題を正解すると次の問題を解答できる . 図 12 にリモートディスプレイ上に表示されるクイズアプリケーションの Web ページを示し , 図 13 にその HTML ソースを示す .

このアプリケーションを提案機構に基づいて実現するために , 共通 ID の名称を “Quiz” , 識別 ID の最大数を “20” , トークンの最大数を “20” とした . また , 提案機構が提供する , “TokenRequest” , “AddDocument” , “IDValidPeriod” , “TokenValidPeriod” , “TokenMethod” , “CompareVariable” , および , “AutoUpdate” 関数を使用した . “AddDocu-



図 8 トークンを持つユーザのための操作ページ
Fig. 8 An operational page for a user with a token.



図 9 トークンを持たないユーザのための操作ページ
Fig. 9 An operational page for a user without a token.

ment” 関数により，正解したトークンを持つユーザのための操作ページには「It is your turn.」を表示し，誤答したトークンを持たないユーザのための操作ページには「Your answer is wrong.」と表示した．“IDValidPeriod” 関数により，共通 ID と識別 ID の有効時間を 30 秒とし，“TokenValidPeriod” 関数により，トークンの有効時間を 25 秒とした．したがって，トークンを持つユーザは 25 秒以内に回答しなければならない．“TokenMethod” 関数により，ユーザが要求する順番にトークンを割り当てる．また，問題番号，ユーザの回答番号，正解番号の変数として，それぞれ，“question”，“answer”，“correct” を使用した．そこで，“CompareVariable” 関数により，各ユーザの“answer” の値と，“correct” の値が等しい場合，つまり，ユーザが正解した場合は，操作ページに「Your answer is right!」と表示する．さらに，“AutoUpdate” 関数により，ユーザの携帯電話上での操作とは無関係に，リモートディスプレイ上の Web ページを自動更新させる．

トークンを持つユーザのための操作ページは，答

```
<HTML><HEAD><TITLE>Othello</TITLE></HEAD>
<BODY>
It is your turn.
<FORM ACTION="http://RDserver.cgi"><HR>
Vertical: <SELECT NAME="vertical">
  <OPTION VALUE="1">1 . . . . </SELECT>
Horizontal: <SELECT NAME="horizontal">
  <OPTION VALUE="1">1 . . . . </SELECT>
<INPUT TYPE="hidden" NAME="color"
  VALUE="white">
<INPUT TYPE="submit" VALUE="--OK--">
<INPUT TYPE="hidden" NAME="comID"
  VALUE="Othello">
<INPUT TYPE="hidden" NAME="perID"
  VALUE="1">
<INPUT TYPE="hidden" NAME="token"
  VALUE="1"></FORM><HR>
<A HREF="http://RDserver.cgi?
  comID=Othello&perID=1&token=1&
  url=http://othello.cgi?exit=1">Exit</A>
</BODY></HTML>
```

図 10 操作ページ（トークンあり）の HTML ソース（抜粋）
Fig. 10 A source code of the operational page for a user with a token.

```
<HTML><HEAD><TITLE>Othello</TITLE></HEAD>
<BODY>
Wait a minute!<HR>
<A HREF="http://RDserver.cgi?
  comID=Othello&perID=2&token=0&
  url=http://othello.cgi">Request</A><HR>
<A HREF="http://RDserver.cgi?
  comID=Othello&perID=2&token=0&
  url=http://othello.cgi?exit=1">Exit</A>
</BODY></HTML>
```

図 11 操作ページ（トークンなし）の HTML ソース（抜粋）
Fig. 11 A source code of the operational page for a user without a token.

えを選択するためのフォームを含む．一方，トークンを持たないユーザのための操作ページは，あらかじめクイズに参加し，トークンを取得するリンクを含む．

4.3 王様ゲーム

王様ゲームは，番号を割り振られた参加者の中から王様をくじなどで決定し，王様が他の参加者に対して何でも命令できるゲームである．王様から指名された参加者は，王様の命令に対して必ず従わなければならない．図 14 にリモートディスプレイに表示される Web ページを示す．また，図 15 にその HTML ソースを示す．

このアプリケーションを提案機構で利用する際，共通 ID の名称を「King」，識別 ID の最大数を「5」，ト

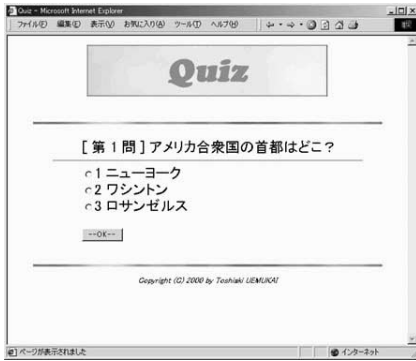


図 12 アプリケーション例: クイズ

Fig. 12 An application example: Quiz.



図 14 アプリケーション例: 王様ゲーム

Fig. 14 An application example: King game.

```

<HTML><HEAD><TITLE>Quiz</TITLE>
<!--<RD comID="Quiz" maxperID="10"
token="10">-->
<!--<RDLIB>TokenRequest::
AddDocument(1,"It is your turn.")::
AddDocument(0,"Your answer is wrong.")::
CompareVariable("answer", "correct",
  "Your answer is right!")::
IDValidPeriod(30)::TokenValidPeriod(25)::
TokenMethod(first)::
AutoUpdate("http://quiz.cgi?q=1", "30")
</RDLIB>-->
</HEAD><BODY>
.....
<FORM ACTION="http://quiz.cgi">
Q1 :Who is the next President of
the United States?<HR>
<INPUT TYPE="radio" NAME="answer"
VALUE="1">1 Bill Clinton<br>
.....
<INPUT TYPE="hidden" NAME="question"
VALUE="1">
<INPUT TYPE="hidden" NAME="correct"
VALUE="2">
<INPUT TYPE="submit" VALUE="--OK--">
</FORM>
</BODY></HTML>

```

図 13 クイズアプリケーションの HTML ソース (抜粋)

Fig. 13 A source code of the Quiz application.

クンの最大数を「1」とする。また、提案機構が提供する、“TokenRequest”、“AddDocument”、“TokenMethod”関数を使用する。“AddDocument”関数により、トークンを持つユーザのための操作用ページには「You are the King!」を表示し、トークンを持たないユーザのための操作用ページには「You must obey the King.」と表示する。“TokenMethod”関数により、トークンは両IDを持つ5人のユーザの中からランダムに決定する。

```

<HTML><HEAD><TITLE>King</TITLE>
<!--<RD comID="King" maxperID="5"
token="1">-->
<!--<RDLIB>TokenRequest::
AddDocument(1,"You are the King!")::
AddDocument(0,"You must obey the King.")::
TokenMethod(random)</RDLIB>-->
</HEAD><BODY>
.....
<FORM ACTION="http://king.cgi">
1番<INPUT TYPE="radio" NAME="sbj" value="1">
.....
<SELECT NAME="select">
.... <OPTION VALUE="2">100円あげる</SELECT>
.....
<TEXTAREA COWS="30" NAME="area">
</TEXTAREA><BR>せよ!
<INPUT TYPE="submit" VALUE="王様は私だ!!">
</FORM>
.....
</BODY></HTML>

```

図 15 王様ゲームアプリケーションの HTML ソース (抜粋)

Fig. 15 A source code of the King game application.

トークンを持つユーザのための操作用ページは、王様として命令を出すためのフォームを含む。一方、トークンを持たないユーザのための操作用ページは、トークンを要求するためのリンクを含む。

以上3つのアプリケーション例の特徴を表1に示す。

5. 考 察

本章では、本論文で提案したアクセス制御機構を中心に考察を行う。

5.1 提案機構の適用範囲

提案機構では、各アプリケーションはHTMLで記述されたWebページとして実現される。そのため、携帯電話やインターネット、Webブラウザなどの既存の

表 1 各アプリケーション例の比較
Table 1 Comparison among applications.

	オセロ	クイズ	王様ゲーム
共通 ID の名称	Othello	Quiz	King
識別 ID の最大数	2	10	5
トークンの最大数	1	10	1
ID の有効時間 (IDValidPeriod)	60	30	60 (default)
トークンの有効時間 (TokenValidPeriod)	45	25	50 (default)
トークンの配布方法 (TokenMethod)	match	first	random
その他のライブラリ関数	TokenRequest, AddDocument	TokenRequest, AddDocument, AutoUpdate, CompareVariable	TokenRequest, AddDocument

インフラストラクチャが利用でき、システム開発に要する費用と時間が少ない。

しかし、今日の携帯電話の WWW ブラウジング機能は、基本的な HTML と CGI にしか対応していないため、提案機構では、ユーザは操作用ページ上のリンクとフォームに対する操作しか行うことができない。したがって、ユーザが行った操作に対して次の操作用ページが携帯電話上に表示されるまで多くの時間がかかってしまう。そのため、利用するアプリケーションには、シューティングゲームなどのリアルタイム性を要するものは適していない。

このような問題を解決するには、まず、今後の携帯電話の性能向上が期待される。現在、IMT-2000 をはじめとして、開発がさかんに進められている次世代携帯電話では、大容量かつ高速な通信や、動画の伝送が可能である。これにより、Web ページの表示時間に対するユーザの不満が解消され、リアルタイム性を要する動的なコンテンツも利用できる。しかし、携帯電話のメモリ容量や消費電力、ディスプレイの大きさなどのハードウェアに関する制限のすべてが解消されるとは考えられない。したがって、今後、携帯電話の性能向上とともに、より多くのアプリケーションをリモートディスプレイ上で利用できるように、提案機構の機能を拡張することを検討している。その 1 つとして、次節で説明する Java への対応があげられる。

5.2 Java への対応

今日、NTT ドコモ⁶⁾が提供する「i アプリ」をはじめとして、携帯電話上で Java アプリケーションを利用できるサービスが普及している。そこで、提案機構を、以下のように Java (i アプリ) に対応させることを検討している。

これまででは、共通 ID と識別 ID を持ち、かつトーク

ンを持っていないユーザがトークンを要求する場合、そのユーザは操作用ページ上に追加されたリンクを選択する必要があった。これは、携帯電話の WWW ブラウジング機能が、HTML の<META>タグの refresh 属性に対応していなかったためである。そこで、i アプリのエージェント機能を利用し、トークンが解放されているかどうか自動的にページ変換サーバに問い合わせる方法を検討している。この方法により、ユーザ間のトークンの交換がスムーズになる。

また、トークンを持つユーザの携帯電話上の操作用ページに表示する内容を、動的に変化させることを検討している。たとえば、オセロのアプリケーションにおいて、トークンを持つユーザが駒の位置を入力している間に、ID の制限時間をカウントダウンする機能を実現することを検討している。これは、i アプリのスタンドアロン型のアプリケーションを利用することにより実現できる。

さらに、リモートディスプレイ上で刻々と変化する表示画面の縮小版を自動作成し、操作用ページに追加する機能を実現することを検討している。これは、リモートディスプレイ上の画面をキャプチャし、携帯電話の画面サイズに縮小したものを、ページ変換サーバが定期的に作成し、それを携帯電話が i アプリのエージェント機能を用いて要求することにより実現できる。

5.3 ユーザの優先順位

提案機構では、リモートディスプレイ環境が、いたるところに設置されたコンピュータをだれでも自由に利用できるというユビキタスコンピューティング環境の性質を持っていることを考慮し、ID (共通 ID と識別 ID) を配布する際のユーザの優先順位はすべて同じものとしている。つまり、ユーザがリモートディスプレイにアクセスしてきた順番に、制限数に達するま

で、各 ID を割り当てる。

しかし、1つの企業内など特定のコミュニティでリモートディスプレイを使用する場合には、ユーザの優先順位を異なるものとした方がよい場合がある。そこで、今後、ユーザの年齢や職業、地位などのプロフィール情報をもとにして、リモートディスプレイにアクセスする優先順位をつけることを検討している。文献4)の WebSplitter 機構では、ユーザの地位(先生と生徒)や、ユーザの所持する機器の特徴などを記述したポリシーファイルをもとに、講義で使用する Web ページを分割し、それぞれの機器に出力する。このポリシーファイルは、Web ページ作成者が XML によって記述する。先生が所有する PDA には、Web ページの操作ボタンと授業に関する注釈を表示されるが、生徒の所有する PDA には、その他の図やテキストのみが表示される。この機構では、講義や会議といったある程度限定した場所やユーザを想定しているが、リモートディスプレイ環境では、リモートディスプレイに対して、不特定多数のユーザがアクセスすることを想定している。そこで、リモートディスプレイ環境において、個人ユーザを識別してアクセス制御を行うには、ページ変換サーバがプロフィール情報を持ち、ユーザがアクセスする際にパスワードなどを入力する方法よりも、あらかじめ各ユーザがプロフィール情報を持ち、アクセスする際にページ変換サーバがその情報を参照する方法が適している。これによって、ユーザの所持するプロフィール情報をもとにアクセス制御を行うだけでなく、各ユーザの嗜好に応じてカスタマイズされた操作用ページを作成できる。

また、提案機構では、TokenMethod 関数によってトークンを取得する優先順位をアプリケーション開発者が指定できる。現段階では、match, random, first の3つのみを指定できるが、今後、さまざまなアプリケーションの利用を考慮し、その他のトークン配布方法を提供することを検討している。

5.4 関連研究

ユビキタスコンピューティング環境におけるアクセス制御に関する研究として、文献4)以外にも文献3)や文献7)がある。文献3)では、家などのユビキタスコンピューティング環境内に遍在するさまざまな情報やサービスへのアクセスに対して、environment role に基づいて制御する手法を提案している。environment role とは、時刻やユーザの位置、状態(忙しさや体調)など、アクセスを許可するためのルールを定義したものである。ユーザがアクセス要求を発生した時点で、その環境が environment role によって定義された条件

を満たしているかどうかを動的に判断する。このシステムでは、environment role によってアクセスを許可する際の状況を詳細に定義し、各ユーザを個別に識別してアクセス制御を行うことができる。一方、提案機構では、リモートディスプレイにアクセスする全ユーザは同じ優先順位を持っており、家やオフィスなどだけでなく、広範囲のコミュニティでも利用できる。

また、文献7)では、多くのモバイルユーザがネットワーク内の大規模なサービスにアクセスする際に、ユーザ個人を識別するのではなく、ユーザがアクセスする権利を持っているかどうかを識別するために、チケットを使用している。これは、通常の公共交通機関などを利用する際のチケットと同じ概念であり、有効期限や QoS の情報が含まれている。ある環境内でチケットを買ったユーザは、どのサービスを利用するかを選択する。この際、チケットが本物のサーバから提供されたものであるかを認証したうえで、ユーザのサービスへのアクセスを許可する。また、受け取ったチケットは、複製や偽造、譲渡に対するセキュリティも保証する。このシステムでは、サービスが提供される環境全体に対するアクセス制御を実現しているが、提案機構では、個々のアプリケーションに対するアクセス制御を実現している。

6. おわりに

本論文では、筆者らが提案しているリモートディスプレイ環境において1つのリモートディスプレイに複数のユーザが同時にアクセスし、協調作業や対戦型ゲームを行うことを考慮した、アクセス制御機構を提案した。さらに、アクセス制御機構を、筆者らがこれまでに実装した WWW ブラウジングシステム上に実装した。提案機構では、共通 ID、識別 ID、および、トークンを利用する。共通 ID により、リモートディスプレイを共有するユーザが利用しているアプリケーションを識別し、識別 ID により、各アプリケーションを利用しているユーザを識別する。さらにトークンにより、両 ID を持つユーザに対してリモートディスプレイを同時に操作する権限を与える。共通 ID と識別 ID を持つユーザ間でトークンをやりとりすることにより、複数のユーザがリモートディスプレイを共有できる。

今後の課題としては、リモートディスプレイ上でさまざまなアプリケーションを容易に構築できるように、ページ変換サーバ上で提供するライブラリを拡張する予定である。

謝辞 本研究は、日本学術振興会未来開拓学術研

究推進事業における研究プロジェクト「マルチメディア・コンテンツの高次処理の研究」(Project No.JSPS-RFTF97P00501), および, 文部科学省科学技術振興調整費研究課題「モバイル環境向 P2P 型情報共有基盤の確立」の研究助成によるものである。

参考文献

- 1) Beigl, M.: Using spatial co-location for coordination in ubiquitous computing environment, *Proc. International Symposium on Handheld and Ubiquitous Computing (HUC'99)*, pp.259–273 (1999).
- 2) Buyukkokten, O., Molina, H.G., Paepcke, A. and Winograd, T.: Power browser: efficient web browsing for PDAs, *Proc. Human-Computer Interaction Conference 2000 (CHI 2000)*, pp.430–437 (2000).
- 3) Covington, M.J., Long, W., Srinivasan, S., Dey, A.K., Ahamad, M. and Abowd, G.D.: Securing context-aware applications using environment roles, *Proc. ACM Symposium on Access Control Models and Technologies (SACMAT'01)*, pp.10–20 (2001).
- 4) Han, R., Perret, V. and Naghshineh, M.: Web-Splitter: A unified XML framework for multi-device collaborative web browsing, *Proc. ACM Conference on Computer Supported Cooperative Work (CSCW'00)*, pp.221–230 (2000).
- 5) Masui, T.: POBox: an efficient text input method for handheld and ubiquitous computers, *Proc. International Symposium on Handheld and Ubiquitous Computing (HUC'99)*, pp.289–300 (1999).
- 6) NTT DoCoMo Net. URL:http://www.nttdocomo.co.jp
- 7) Patel, B. and Crowcroft, J.: Ticket based service access for the mobile user, *Proc. International Conference on Mobile Computing and Networking (Mobicom'97)*, pp.223–233 (1997).
- 8) Rekimoto, J.: A multiple-device approach for supporting whiteboard-based interactions, *Proc. ACM Conference on Human Factors in Computing Systems (CHI'98)*, pp.344–351 (1998).
- 9) Robertson, S., Wharton, C., Ashworth, C. and Franzke, M.: Dual device user interface design: PDAs and interactive television, *Proc. ACM Conference on Human Factors in Computing Systems (CHI'96)*, pp.79–86 (1996).
- 10) Schmidt, A., Beigl, M. and Gellersen, H.: There is more to context than location, *Proc. Interactive Applications of Mobile Computing (IMC)*, pp.11–20 (1998).
- 11) 塚本昌彦: モバイルコンピューティング, 岩波科学ライブラリー, 岩波書店 (2000).
- 12) The LWP homepage. URL:http://www.linpro.no/lwp/
- 13) 上向俊晃, 萩野浩明, 原隆浩, 塚本昌彦, 西尾章治郎: リモートディスプレイ環境における WWW ブラウジングシステム, 情報処理学会論文誌, Vol.41, No.9, pp.2364–2373 (2000).
- 14) Weiser, M.: The computer for the twenty-first century, *Scientific American*, Vol.265, No.3, pp.94–104 (1991).

(平成 13 年 5 月 29 日受付)

(平成 13 年 12 月 18 日採録)



上向 俊晃

2000 年大阪大学工学部情報システム工学科卒業。2001 年同大学院工学研究科博士前期課程修了。現在, 同大学院工学研究科博士後期課程在学中。モバイルコンピューティング, コピキタスコンピューティングに興味を持つ。



原 隆浩 (正会員)

1995 年大阪大学工学部情報システム工学科卒業。1997 年同大学院工学研究科博士前期課程修了。同年同大学院工学研究科博士後期課程中退後, 同大学院工学研究科情報システム工学専攻助手となり, 現在に至る。工学博士。1996 年本学会山下記念研究賞受賞。2000 年電気通信普及財団テレコムシステム技術賞受賞。データベースシステム, 分散処理に興味を持つ。IEEE, 電子情報通信学会の各会員。



塚本 昌彦 (正会員)

1987 年京都大学工学部数理工学科卒業。1989 年同大学院工学研究科修士課程修了。同年, シャープ(株)入社。1995 年大阪大学大学院工学研究科情報システム工学専攻講師, 1996 年より, 同大学院工学研究科情報システム工学専攻助教授, 現在に至る。工学博士。時空間データベースおよびモバイルコンピューティングに興味を持つ。ACM, IEEE 等 7 学会の会員。



萩野 浩明(学生会員)

1996年大阪大学工学部情報システム工学科卒業。1998年同大学院工学研究科博士前期課程修了。2001年同大学院工学研究科博士後期課程修了。同年(株)NTTドコモ入社、現在に至る。工学博士。モバイルコンピューティング、知識処理に興味を持つ。



西尾章治郎(正会員)

1975年京都大学工学部数理工学科卒業。1980年同大学工学研究科博士課程修了。工学博士。京都大学工学部助手、大阪大学基礎工学部および情報処理教育センター助教授を経て、1992年より大阪大学大学院工学研究科情報システム工学専攻教授となり、現在に至る。2000年より大阪大学サイバーメディアセンター長を併任。この間、カナダ・ウォータールー大学、ビクトリア大学客員。データベース、知識ベース、分散システムの研究に従事。現在、Data & Knowledge Engineering, DataMining and Knowledge Discovery, The VLDB Journal等の論文誌編集委員。ACM, IEEE等8学会の会員。