ユビキタス情報機器の操作履歴に基づくユーザ支援

宮崎 俊和† 河口 信夫‡ 稲垣 康善*

†名古屋大学大学院工学研究科

‡名古屋大学情報連携基盤センター *愛知県立大学情報科学部

概要

近年，様々な場所に情報機器が埋め込まれ，多様なサービスを実現するユビキタスコンピューティング環境が整いつつある．一般に機器が提供するサービスを利用するためには，何らかのユーザインタフェースの操作が必要になる．機器が多種多様になることにより，提供されるサービスも多くなるが，必要となる操作も同時に増大する．ユーザに優しい真のユビキタスコンピューティング環境を実現するためには，可能な限りユーザの操作を最小化することが望ましい．

本稿では，多様なユビキタス情報機器の利用履歴を用いて，ユーザの機器利用における操作の予測やマクロ化を行い，ユーザ支援を行う手法を提案する．本手法では，個々の機器の操作が行われる度にその操作内容とそのときの環境情報をネットワークを通じてデータベース化する．このデータベースによってユーザの操作状況が記録される．データベースから状況の類似した一連の操作を検索し，その結果から次の操作の予測を行う．また，複数の機器に対する一連の操作指示を記憶し，単純な指示でそれを実行できる．本手法を実際のユビキタスコンピューティング環境上に実現し，その有効性を確認した．

1 はじめに

近年，様々な場所に情報機器が埋め込まれ，多様なサービスを実現するユビキタスコンピューティング環境が整いつつある．スマートルームのような様々なセンサやコンピュータを備え付けた部屋の研究が多く行われ [1][2]，また屋外では駅構内にホットスポットを設置したり [3]，携帯電話や PHS などの基地局，GPS の位置情報等を利用したセンサネットワークの研究が盛んに行われている [4]．このような環境下で，あるユーザが様々なサービスを利用しようとする場合，ネットワークの接続の方法や場所，またそのときに利用する機器は状況によって異なる．あるネットワークに接続された機器を別のネットワークに接続し直したり，使用する機器のバリエーションから複数のユーザインタフェースを扱ったりと面倒な操作を日常的に繰り返し行わなければならないことが少なくない．ユビキタスコンピューティング環境を実現するためには，このような煩わしい操作をユーザに感じさせないことが必要である．

これに対して，与えられた状況においてユーザに提供されるサービスの設定や，それに伴う煩わしい操作を，一連の手順として簡便に実行できるようにシステムに組み込んでおくという方法もある．しかし，多種多様で特定の状況でのみ行われるこれらの操作をすべてシステムで事前に用意しておくことは不可能である．このような一般的でない煩わしい機械的作業を効率よく実行するためには，あらゆる場合を想定してあらかじめシステムに用意しておくよりも，状況に応じてユーザが次の操作を予測したり，以前の実行内容を再利用することがより有効的であると考えられている [5]．

本稿では，ユーザインタフェースを持つ機器がユーザの操作の履歴とその状況から学習することにより，現在の状況においてユーザが行うであろう操作列を予測し，またその操作列を簡単に実行するシステムを提案する．図 1 はその例であり，ミーティングという状況から簡単な操作でスライドの表示や資料の配布等の操作列を予測して実行する．ユビキタス情報機器は「場所」「位置」「ネットワークのメンバ」「ネットワーク種」等の環境の情報を認識でき，ユーザがある状況でどんな操作をしたかをネットワークを通じて逐次データベース化する．次に，現在の状況と同じかまたは似た状況の操作履歴から次の操作を予測しユーザに提示する．また最近の履歴を用いることにより，ある一連の操作指示を簡単な操作で行うための設定を支援し，簡単な操作や状況の変化をトリガーとしてその一連の操作を実行できる．

本稿の構成は以下の通りである．第 2 節ではユビキタスコンピューティングにおける繰り返し操作の問題点をあげ検討する．第 3 節では第 2 節を踏まえユーザ支援に有効な設計手法を提案する．第 4 節ではその手法を用いたシステムの実装について述べ

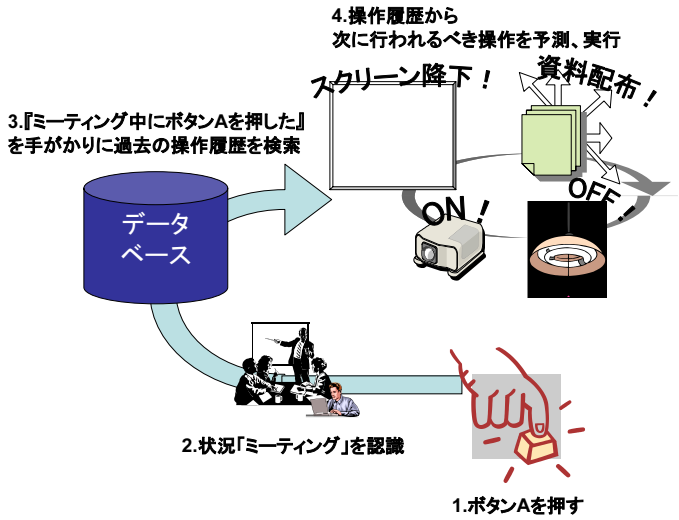

図 1: 簡単な操作での実行

る．第５節で本稿をまとめ今後の課題を示す．

2 ユビキタスコンピューティング環境におけるユーザ支援に関する検討

近年，携帯端末の小型・高性能化，ネットワークの発達によりユビキタスコンピューティング環境が整いつつある．それに伴い，そのような環境で動作する様々なソフトウェア・ミドルウェアが研究されている．しかし，それらを日常的に使用するには様々な問題がある．本節ではその問題を明確にし，操作履歴を利用した有効なユーザ支援の方法を提案する．

2.1 煩雑な繰り返し操作の発生

ユビキタスコンピューティング環境では様々な状況で多種多様な目的から情報機器を使用する．ミーティングや会議ではプロジェクタを利用したプレゼンテーションを行い，議題の資料や前回の議事録を参加しているメンバに配る等の操作を行うであろう．また個人で端末を利用する場合でも，例えばバス停や駅などのホットスポットで時刻表を表示したり，待ち時間をつぶすために，簡単なゲームや web アクセス等を日常的に行うこともあろう．このように将来的に情報機器の使用頻度の増加を考えた場合，ユーザは日常的に繰り返し操作を行うことが予想される．目的を達成するための操作が非常に単純であったり，ネットワークが唯一であったり等の状況の複雑さがないならば，それはまれに行われる少し面倒な操作というだけである．しかし，プレゼンテーションを伴うミーティングが情報機器の揃った場所で毎週行われることを考えたとき，ユーザは「プロジェクタの電源を On にし」「自端末にプロジェクタをつなげ」,「スクリーンを降ろして」「照明を暗くする」．そして，やっと「スクリーンに発表資料を表示し」て，ミーティングのメンバには「電子的な資料を配り」自分は意見やコメントのメモをとるような「テキストエディタを立ち上げる」というような操作を毎回繰り返すことになる．これでは，ユーザはコンピュータを使用することにうんざりしてしまう．

このような環境でユーザに煩雑な操作を強要してしまう原因には，以下の三つが考えられる．

- インタフェースの氾濫
- コンピュータの使用頻度の増加
- 情報機器の連携操作の多様性

一つ目は各ユビキタス情報機器のボタンやスイッチ等のユーザインタフェースの増加である．情報機器が増えれば増えるだけ，そのためのインタフェースが必要になる．二つ目は将来的に更にコンピュータの使用が増加し，それに伴いユーザの日常的な繰り返し操作が多数行われることになることである．三つ目は情報機器の連携が複雑になることである．多様な連携に伴う設定にユーザは煩わしさを感じることになる．

このようなある一つの目的のために行われる行動の煩雑さは人と人の間や社会にも存在する．これらの場合，その作業をマニュアル化したり，まとめることができる作業を一括して一人の人間に任せたりして，簡略化をはかる．また，人間同士の間では相手の現在の状況を認識することによって，簡単な合図で，相手の目的を解し目的達成のための行動を取ることができる．例えば行きつけの定食屋やラーメン店でいつもの料理を頼めば，主人が彼の嗜好を考慮して食べれないものを除き，好むものをおまけしてくれるという事もあるだろう．これは，彼が店の主人に数回細かな要求をした結果であり，それにより彼はいちいち細かい注文をする煩わしさから開放された．多数の情報機器を利用する環境で煩雑な操作を解消することを考えたとき，定食屋であれば細かなメニューまで用意しておくような，最初からシステムにその操作列を用意するという方法ではユーザの多種多様な目的の全てに対応することはできない．後者の，経験的，つまりユーザの操作履歴と状況から意図する操作を予測したり，またユーザが例を示すことによってその例をコンピュータが覚えておき簡単な操作（以後トリガと呼ぶ）で再生できるようなシステムが望ましいと考えられる．このようなシステムは予測インタフェースや例示プログラミングの技術を利用して実現できる．

2.2 予測／例示

予測や例示によるユーザの操作の支援を行う場合，単に予測や例示が可能であるだけでなく，ユーザがその機能を便利だと感じる必要がある．予測された操作がユーザの意図しないものであったり，例示によるプログラム作成が複雑であったりすれば予

測や例示のシステムを利用する意味がなくなってしまう．そこで，予測，例示プログラミングのシステムは次の事を満たす必要がある．

a. プログラム作成が簡単である

例示プログラミングのプログラム作成において，その操作が煩雑であるとプログラム作成における手間の方が作らないときの手間に比べて大きくなってしまう．そのためにはユーザが行った操作の履歴といった暗黙的例示からユーザが意識しなくてもプログラムを生成できる事が望ましい．

b. プログラム実行に対するリスクが少ない

完全にシステムに実行をまかせてしまうと，ユーザが気づかないうちに間違った実行を行ってしまうなどのリスクが生じる．このようなリスクを軽減するために，実行中の操作列を停止したり，場合によっては前の状態に戻るような Undo 機能が必要である．

c. 予測や例示の機能が邪魔にならない

予測結果が常に提示されるようなシステムでは一部のユーザや，またそれを利用しないときにそのような機能が煩わしく感じることがある．また常に予測の機能を動作させておくと予測のやり方によっては機器の処理能力の低下につながることもある．よって，予測や例示はユーザが指示したときに行われるか，行われてもユーザの邪魔にならず，また他の処理に影響を及ぼさない方法が必要である．

3 操作履歴に基づくユーザ支援手法

本節では多数の情報機器を利用する環境で煩雑な操作を解消するためのユーザ支援手法を提案する．前節でこのようなシステムには過去の操作履歴から次の操作を予測したり，面倒な操作を容易に実行するために一連の操作をマクロ化する機能が必要であることはすでに述べた．ここでいうマクロとは，ユーザが指定した一連の操作を順に実行していくプログラムのことである．ユーザの支援の手法は，分散コンピューティング環境における操作履歴の保存方法，操作履歴に基づく操作の予測方法，例示プログラミングを用いたマクロ作成方法の三つに分けられる．

3.1 分散コンピューティング環境における操作履歴の保存

ユビキタスコンピューティング環境では様々な情報機器が分散して存在する．そのため全ての情報機器の操作履歴を特定のデータベースに保存するにはネットワークを介す必要がある．ここではどのように分散した情報機器から操作履歴を収拾し，利用するかを述べる．

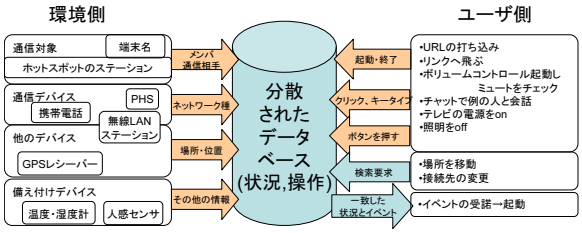

図 2: 操作履歴の保存と利用

3.1.1 操作履歴の内容

図 2 のように履歴情報は「状況」と「操作」の組で保存される.「状況」とはユーザや情報機器を取り囲む環境である．時間や場所，その他の様々な要素が考えられる．例えば「午後，会議室で」でという状況には「時間」「場所」,「研究室の自分の机に座り，無線 LAN に接続して」には「場所」「ネットワーク種」,「温度 10 ℃，湿度 45 %，北緯 35 ° 43’26”、東経 137 ° 02’25” の位置で」では「温度」「湿度」「位置」という要素によって構成される．このような状況を構成する要素は表 1 のような情報の発信源を持つ． これに対して,「操作」とはユーザが能動的に

表 1: 状況の構成要素とそのソース

ソース	状況の構成要素
自端末	起動中のアプリケーション 時間 (年月日、曜日)
通信対象	メンバ (端末名) 通信相手 (人名)
無線 LAN 等のステーション	場所 (××駅前、××研究室)
GPS	位置 (経緯度)
通信デバイス	ネットワーク種 (ダイアルアップ、 有線/無線 LAN)
温度センサ	温度
湿度センサ	湿度
人感センサ	人の有無

行った情報機器に対するふるまいである．例えば，「電灯を点ける」「プロジェクタでスライドを表示する」「OK ボタンを押す」等がある．操作はできるだけ具体的に与え，それを複数組み合わせることによってユーザの意図を満たすことができる．ユーザの意図と操作の例を表 2 に示す．

3.1.2 各情報機器の操作履歴の保存

ほとんどのユビキタス情報機器はサービスをスマートに提供するために，外部から制御することが可能である．そのためにユビキタス情報機器は外部にコントローラを持つことが多い．このコントローラは制御情報をネットワークを介して目的の情報機

表 2: ユーザの意図と操作

意図	具体的な操作
暗く/明るく	照明スイッチの ON/OFF
ウェブサイトへのアクセス	URL の打ち込み →「ok」ボタンを押す
マナーモード	ボリュームコントロール起動 →ミュートをチェック
プレゼンテーション	プロジェクター ON →スクリーン降下 →自端末とプロジェクタ接続 →照明スイッチ OFF →ファイル転送ソフト起動 →資料配布

器に送信するが，本手法ではその制御情報を同時にデータベースに向けて送信し，これを保存する．これによりほとんどの情報機器の操作履歴を逐次保存することができる．また本手法を実装したシステムが各コントローラを制御することで，保存されているものと同様の操作を再生することができる．

3.1.3 操作履歴の区別

本手法では，スマートルームのような部屋に備え付けられているデータベースと、ユーザ個人の情報機器に付属されているデータベースの二つを想定する．後者は前者のようなデータベースの付属するユーザ個人の情報機器から検索要求を出し，データを取得することができる．これにより部屋に埋め込まれたり備え付けられている情報機器の利用履歴やマクロのデータを初めてその場所に来たユーザがすぐに利用することができる．一方，ユーザのプロフィールに特化したデータを個人の情報機器のデータベースに保存することで，そのユーザのみ必要とする操作や操作を予測，再生できる．

3.2 操作履歴に基づいた次の操作の予測

予測は操作履歴を基に行われる．操作履歴を保存すると同時にそのときのユーザの置かれている「状況」を一緒に保存する．このとき「状況」は「場所」や「時間」といった様々な要素から構成される集合として表現される．予測はまず操作履歴を時間的にまとまった操作列に分けることから始まる．ここで操作列とはある意図をもった一連の操作を表す操作のリストである．次にその操作列から予測を指示したときの状況と同じかまたは似た状況のものを検索する．ここで同じかまたは似た状況の過去の操作履歴を得るために，現在の状況をクエリとしてデータベースを検索する．次に検索された操作列の集合の中に直前に行われた操作列を含むものを探し，あればその操作列から直前に行われた操作列の次の操作にあたる操作を取り出し，予測結果として提示する．

しかしこの方法だけではユーザの期待と異なる予測をする可能性がある．そこで答えを一つに絞らずに複数の予測結果を提示し，ユーザに決定を委ねることとする．これにより間違った予測のリスクを軽減することができる．また，提示されている操作がユーザの求めるものでは無かった場合，ユーザは他の予測を見ることができる．このとき一番もっともらしい予測から提示することが望ましい．ここでいうもっともらしいとは，時間的に最近のものであり，またユーザや端末の置かれている状況がより近いものとする．予測はユーザが指示したときに開始されるので，ユーザは予測機能を煩わしく感じることはない．

3.3 例示プログラミングによるマクロ作成

ユーザがマクロの作成を指示すると直前のまとまった操作を提示する．直前のまとまった操作は上記の予測と同じ方法で操作列として検出される．ユーザがマクロの作成を決定することにより操作列をマクロとして覚えさせることができる．ユーザがあらかじめ例示プログラミングの開始を指定する必要がなく，操作履歴からマクロ化したい操作列を提示できるので，マクロの作成は比較的簡単である．ここでもし提示した操作列がユーザの意図に反する（つまり開始位置がユーザの意図と異なる）場合，操作列のまとめ方を変え再度提示しなおす機能も必要となる．

設定されたマクロを実行すると，定義された操作列が実行される．マクロはいくつも定義することができ，また複数のマクロを同じインタフェースを設定しても状況によって実行するマクロを区別することができる．

4 ユーザ支援システムの設計

本節では前節で述べた手法に基づいて，多数の情報機器を利用する環境で煩雑な操作を解消するためのユーザ支援のシステムを設計する．

4.1 履歴データ

前節で述べたように履歴データは「状況」と「操作」の組で成り立つ．そこで履歴データを l とし状況を s，操作を a としたとき、$l = (s, a)$ と定義する．また状況 s は様々な要素で構成され，操作 a は操作名 n と操作対象 t とその他の属性 w からなる．ある履歴データ l の構造は図 3 のようになる．

ある意図をもった操作列を一つの塊として保存すると，その再現は簡単になるが，履歴情報の応用範囲を狭めることになる．履歴はできるだけプリミティブな形で保存することが望ましい．一方，ある

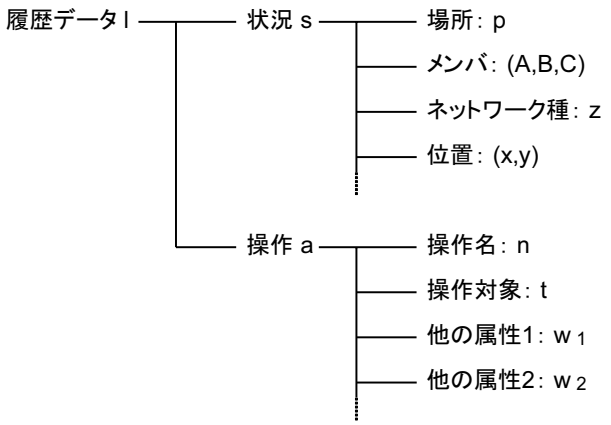

図 3: 履歴データ

意図をもった操作列を一つの塊として保存する事もマクロを記憶しておくという意味で必要となる．そこで，例示プログラミングによって作成されたプログラム，いわゆるマクロもデータベースに保存する．そのデータはトリガとなるボタンや状況の変化に関連付けられていて，トリガが発生したときに検索され，利用される．マクロとして保存されるデータは履歴データの「操作」の部分の集合で構成される．

4.2 操作履歴の保管場所

本手法の履歴データは場所や空間に依存するものとパーソナルなものに分けられる。パーソナルなデータはユーザ個人の情報端末に内蔵されたデータベースか．もしくは広域なネットワークを経て特定のデータベースサーバから情報を得ることができる．このようなデータベースを利用し，場所や空間に依存するようなデータも特定のデータベースに保存することで，必要なときに遠く離れたデータベースからその情報を取得する方法もある．しかし，真のユビキタスコンピューティングではユーザがパーソナルな情報端末を持たず，離れたデータベースからそのユーザ特有のデータを取得できないこともあるし，持っていても広域的なネットワークを利用せずにアドホックなネットワークのみを利用することもある．このように環境では，常に唯一のデータベースに繋がる環境を維持することは難しいと考えられる．

よって本手法では，ユーザ個人の情報端末にデータベースを置くとともに，特定の場所や空間にデータベースを置くような分散的なデータベースが適している．

4.3 インタフェースと予測及びマクロ作成の手順

操作予測や例示プログラミングの機能に必要となるインタフェースとして,「予測ボタン」「ブランクボタン（マクロボタン)」「実行ボタン」の三つを定義する．

予測ボタン

予測ボタンを押すと次のユーザの操作を予測し，提示する．更に予測ボタンを押すと別の予測が提示される．

ブランクボタン（マクロボタン）

ブランクボタンはマクロを定義されるボタンで，マクロの定義されたブランクボタンはマクロボタンとなる．ブランクボタンを押すとマクロ候補が提示され，更にブランクボタンを押すと別の候補が表示される．マクロボタンを押すとマクロが実行される．マクロボタンを一定時間が押しつづけてから離すとマクロが削除され，ブランクボタンになる．

実行ボタン

実行ボタンは予測された操作を実行する．またマクロ作成を実行する．

各ボタンの関係と押されたときの機能を以下に示す.

- 予測ボタン「予測の提示」
  →（予測ボタン「別の予測を提示」）
  →実行ボタン「予測された操作の実行」
  →ブランクボタン「予測して実行された操作の提示」
  →実行ボタン「予測して実行された操作をマクロとして定義」
- ブランクボタン「マクロの候補提示」
  →（ブランクボタン「別のマクロの候補を提示」）
  →実行ボタン「マクロの定義」
- マクロボタン「マクロの再生」
  →実行ボタン「マクロの消去」
- 実行ボタン「操作，操作列の実行中にそれを停止」

4.4 操作列の決定規則

操作列は，時間的に連続性のある操作を関連性があるとしてまとめたものである．ある時間を決め、操作と操作の発生時刻の間隔がそれより長いか短いかで時間的な関連性を判断する．決められた時間より短かければその操作は関連性があるとし操作列としてまとめる．以下に操作列を決定するアルゴリズムを示す．ここで a_i は操作であり，a_{i+1} は a_i の次の操作とする．また $i = 0, 1, 2, 3 \ldots n-1$ であり，n は操作の数である．$t(a_i)$ を操作 a_i の発生した時刻とする．

1. $i = 0$，$j = 0$，操作列 $L_j = [a_i]$
2. $\Delta t = t(a_{i+1}) - t(a_i)$
3. $\Delta t > \theta$ ならば，$j++$
4. $L_j = [a_{i+1}|L_j]$
5. $i++$，$i < (n-1)$ ならば 2 へ

4.5 予測のアルゴリズム

前小節で述べた方法で作成された操作列の集合を基に，次の操作列を予測するためのアルゴリズムを述べる．基本的な方針は直前の操作列と同じ状況の似た操作列を操作列の集合から捜し出すことにある．ここで状況はそれを構成する要素の集合で表現され，例えば「ネットワークのメンバ」と「場所」と「温度」を要素とする状況 s を定義すると，$s = \{(\text{"}Member\text{"}, \{\text{"}NodeA\text{"}, \text{"}NodeB\text{"}, \text{"}NodeC\text{"}\}), (\text{"場所"}, \text{"}LAB\text{"}), (\text{"温度"}, 27.01)\}$ というようになる．直前の操作列 $L_l = [a_h a_{h+1} a_{h+2} \dots a_{n-1}]$ とする．また操作 a_i の行われた状況を $s(a_i)$ で表す．操作列の状況はその先頭の操作の状況であり，状況 $s(L_j) = s([a_k a_{k+1} a_{k+2} \dots]) = s(a_k)$ とする．

1. $j = l$，$s_l = s(L_l)$
2. $j--$，$j < 0$ ならば 8 へ，$s(L_j) \neq s_l$ ならば 2 へ
3. $L_j = [b_0 b_1 b_2 \dots b_{m-1}]$ とし，また $i = 0$
4. $(n - 1 - h) \geq m$ ならば，2 へ
5. $a_{i+h} \neq b_i$ ならば 2 へ
6. $i++$，$(i + h) < n$ ならば 5 へ
7. $b_i b_{i+1} \dots b_m$ をユーザが行う次の操作として提示
8. 目的の操作が得られない場合，
   $s_l := ambiguous(s_l)$ についてこのアルゴリズムを行う（ここで $ambiguous(s_l)$ は s_l よりも状況が曖昧な状況を表す）

例えば，操作として α,β,γ,δ とし，簡単のため状況をネットワークのメンバの集合として考え M, A, B, C とし，履歴データが表 3 のようにあるとする．また操作列 No の若い操作列がより最近のものとする．

表 3: 履歴情報の例

操作列 No	状況 (メンバ)s	操作列 a
操作列 0	M, A, B	$\alpha\beta$
操作列 1	M, A, B	$\alpha\delta\beta$
操作列 2	M, A, C	$\alpha\beta\delta$
操作列 3	M, A, B	$\alpha\beta\gamma$
操作列 4	M, A, B, C	$\alpha\delta$
操作列 5	M, B, C	$\beta\gamma$
操作列 6	M, A, B	$\alpha\beta\gamma\delta$

今操作列 0 が直前の操作列の情報で，$\alpha\beta$ の次の操作を γ と予測したい．上記のアルゴリズムの 2 より，操作列 1，3，4，6 が抽出される．その中から最も最近の操作列 3 から，アルゴリズムの 3，4，5，6，7 で三つめの γ が予測結果として提示されることになる．更に γ の実行を決定し，予測を続けるならば次に操作列 6 の δ も予測することができる．もし γ が求める操作でなければ，アルゴリズムの 8 より例えばメンバ A を削った状況に対してアルゴリズムを適用し，L' は操作列 1,2,3,4,6 となり，結果として操作列 1 から δ が予測される．

5 SmartMacro システムの実装

前節までにユビキタスコンピューティング環境における煩雑な操作の存在と予測や例示によるユーザ支援に関する問題をあげ，有効なシステムの設計手法を提案した．本節ではその手法にしたがったシステムをモバイルエージェントシステム Cogma(Cooperative Gadgets for Mobile Appliances)[2] に実装する．

5.1 実装環境

今回の実装では状況の構成要素として，ネットワークのメンバに注目する．ネットワークのメンバの値は端末名を利用する．ユーザの操作は主にアプリケーションの起動と終了，Java の Button や Menu に代表される GUI を対象とする．

Cogma は JAVA 言語で設計されており，ユビキタスコンピューティング環境における情報機器の連携アプリケーションを容易に実現する目的で作成されたミドルウェアである．Cogma を利用することによりプロジェクタ，照明，スクリーンなどをアドホックネットワークを介して容易に操作できる．

5.2 履歴データベース

履歴情報を保存するデータベースは Xindice という XML データベースを用いる．今回のような場合，状況を構成する要素によってその情報の構造が異なっているため一定の型を持つデータ構造では扱いきれない．また将来的に全く新しいタイプの情報を，状況の構成要素として後から付加することを考えると，拡張性の高いデータ構造が望ましい．XML データベースは異なる文書構造をデータベースに保管し，それらをまたいで照会することができるので，その点で有利であるといえる．以下に今回使用する履歴データの構造を示す．

```
<data date="20xx/xx/xx xx:xx:xx"
            milidate="ミリ秒">
 <action name="操作名"
  target="アプリケーション１の名前:ID"
  属性名１="属性値１" 属性名２="属性値２">
 </action>
 <status>
  <member num="2">
   <mynode name="自端末の名前" />
   <node name="他端末１の名前" />
  </member>
```

```
  <registered num="2">
   <codget
    name="アプリケーション１の名前と場所">
    アプリケーション１の名前:ID
   </codget>
   <codget
    name="アプリケーション２の名前と場所">
    アプリケーション２の名前:ID
   </codget>
  </registered>
 </status>
</data>
```

一つの履歴情報 data は action タグと status タグからなる．また属性として，時刻の情報をもつ．

action タグは操作の情報のタグである．name という属性は操作名を表し，target という属性は操作の対称となるアプリケーションの名前と ID を表す．また他の属性はアプリケーションの作成者が自由に設定できる．ユーザの能動的な操作でなく，状況の変化に伴うイベントは同じようにして action タグの代わりに change タグというタグで書き込まれる．status タグは状況を構成する要素のタグを持つ．上の例ではネットワークのメンバをあらわす member タグと現在起動されているアプリケーションを表す registered タグがある．基本的に各状況の構成要素のタグはそれぞれ異なる構造を持っている．member タグではネットワークのメンバの数を属性として持ち，mynode タグと node タグを持つ．この二つのタグは端末名を表す属性を持っている．registered タグもほぼ同様である．

またこのような履歴情報をユビキタスコンピューティング環境でどこに置くかという問題に関して前述したが，今回は簡単のために個人の端末にそれぞれデータベースがあるものとする．

5.3 モジュールの構成と実装

次に本システムの構成を図 4 に示す．実装は「各アプリケーションの操作や状況の変化の記録を行う」モジュールと「状況を監視し，予測やマクロの作成，再生を行う」モジュールが中心となる．前者は履歴保存モジュール，後者は予測/例示モジュールと定義する．これらの二つのモジュールはネットワークのメンバの追加や離脱といった状況の変化を感知しそれに対応する必要がある．Cogma ではその機能を LinkMonitor と呼ばれるインターフェースで提供している．LinkMonitor を実装することで上記の二つのモジュールは，端末の追加と離脱を管理する LinkManager から通知される．

また一つの操作を抽象化したクラスが必要となる．履歴情報の XML リソースからこのクラスのインスタンスを作成し，またこのクラスのインスタンスをアプリケーションに渡すことで具体的な操作を実行する機能が必要である．

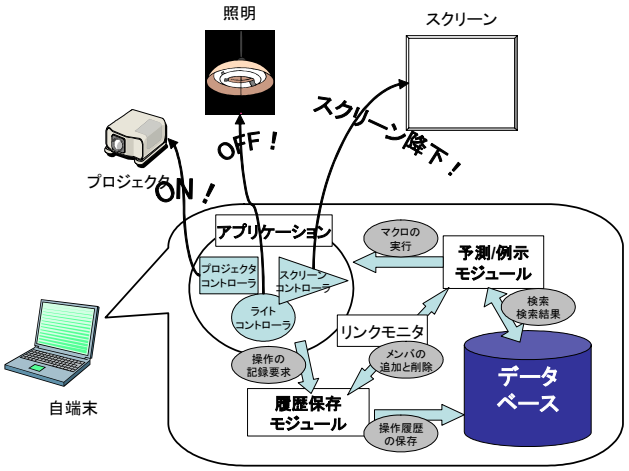

図 4: システム構成図

本システムを大まかに捉えると，図 4 のように [各アプリケーション]—[履歴保存のモジュール，予測/例示モジュール]—[データベース] の間を単一操作を抽象化したクラスが行き来する構成となっている．

以下に各モジュールの機能を示す．

履歴保存モジュール

状況を監視し，イベントがある毎に保存．この場合イベントとは，オンラインメンバの追加と離脱，システムに対するアプリケーションの登録と解除，システムに登録されているアプリケーションの指定された操作である．本システムを実装したアプリケーションはデータベースに登録したい操作の詳細を記述しておかなければならない．操作は action と呼ばれ，target と他の属性を持つ．target は操作の対称となるアプリケーションの名前と ID である．属性は複数もつことが許されており，属性名と属性値の組で構成され，アプリケーション作成者が自由に決定できる．またそれと同時に状況を構成する要素としてネットワークのメンバや現在登録されているアプリケーションの情報も保存する．

予測/例示モジュール

ユーザが予測やマクロの作成を指示したときに，データベースを検索し，予測やマクロの作成を行う．

作成されたマクロを記憶し，そのトリガが引かれたとき対象となっているアプリケーションを操作する．操作は対象となっているアプリケーションの操作再生を行うメソッドを呼び出すことによって行われる．本システムを実装したアプリケーションにはそのメソッドを実装し，呼び出されたときの処理を書かなければならない．またこのメソッドの引数には操作を抽象化したクラスを用いる．また予測された操作の実行も同じように行われる．

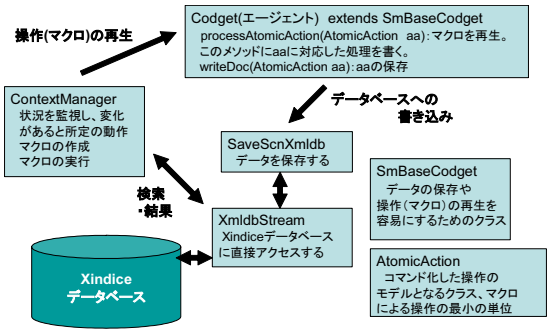

図 5: クラス関係図

操作の抽象クラス

一つの操作を抽象化したクラス．様々な操作をこのクラスで表現することができ，またデータベースから得た履歴情報からこのクラスを一つの操作として容易に構成できる．以下の要素で構成される．

target アプリケーション名とその ID

action 操作名

attributes 属性名と属性値の組のリスト．操作に付随するオプションを示す．

履歴保存モジュール，予測/例示モジュールを SaveScnXmldb，ContextManager というクラスに実装した．図 5 はそれらのクラスとその他の補助的なクラスの関係図である．

また XML データベースの検索には XPath を用いる．オンラインメンバが TSUKSI（自端末）と Kblue の状況となっている履歴情報を検索するときは

```
/data[status[member[@num="2"]
[mynode="TSUKUSI"][node="Kblue"]]]
```

となる．このようなクエリを用い，履歴から同じ状況を探す．

図 6 は実装したシステムのスクリーンショットである．ライトコントローラとライトエミュレータが二つ，またプロジェクタのコントローラ，マトリクススイッチャの操作を記録することができる．また図左上にある exec ボタンや guess ボタンで各操作を予測，再生することができる．

6 おわりに

本稿ではユビキタスコンピューティング環境における日常的な繰返し操作の煩わしさを問題にとりあげた．またそれを解消するための手法としてユビキタス情報機器の操作履歴に基づく予測インタフェースや例示プログラミングを提案し，それをモバイルエージェントシステム Cogma に実装した．

実装では状況の構成要素としてネットワークのメンバしか考慮しなかったため，予測のためのデータベースの検索は容易であった．実際には状況を構成する要素は様々であり，それによって正確な予測や

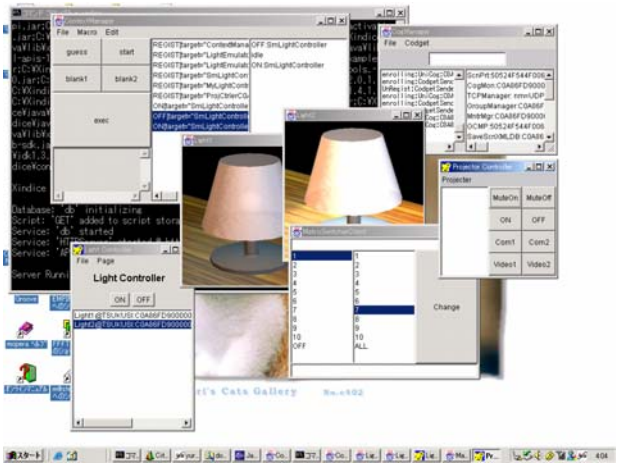

図 6: 実装したシステム

細かいトリガの指定などができるが，データベースの検索は複雑になる．この場合効率の良い検索や予測の手法を検討しなければならない．

また提案した手法で実装したシステムによりユーザ端末から複数の情報機器を利用する一連の操作を簡単な操作で実行すること可能となった．今後はこのシステムを利用し本手法の有効性を評価する必要がある．

参考文献

[1] Microsoft, Easy Living, http://research.microsoft.com/easyliving/

[2] 河口信夫, 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, Nov.2001.

[3] 中村嘉志, 伊藤日出男, 西村拓一, 山本吉伸, 中島秀之, ”無電源小型通信端末 CoBIT による近距離情報支援の実現”, 情報処理学会知的都市基盤研究グループ研究報告, 2002-ICII-3, pp.1-7 (2002).

[4] 岩谷晶子，西尾信彦，村瀬正名，徳田英幸，”ごましお：アドホックセンサネットワークにおけるノード位置決定方式”，情報処理学会モバイルコンピューティングとワイヤレス通信研究会 Vol.2001(108)，2001 年 11 月，pp.23-30

[5] 増井俊之, ”予測/例示インタフェースシステムの研究”, 学位論文, 東京大学, November.1997

[6] Henry Lieberman, ”Your Wish Is My Command Programming by Example”, Morgan Kaufmann Publishers , 2001, 416p

[7] 古市悠, 志和木愛子, 岩井将之, 徳田英幸, ”Sticky Editor System:基盤型実世界指向アプリケーション”, マルチメディア, 分散, 強調とモバイル (DICOMO2003) シンポジウム論文集, pp.649-652, June.2003