

知識メディア技術を用いたプロアクティブ環境フレームワーク

中山元也[†] 猪村元[†] 田中譲[†]

近年、ウェブ関連技術、センサ・ネットワーク技術の発展により、ユーザの周辺環境の状況などがウェブを介して容易に取得可能となった。これらの情報や既存のウェブアプリケーションを連携させることで構築できる環境システムへの需要は高い。本稿では ECA(Event-Condition-Action) ルールを用いてウェブ・コンテンツ間の連携を定義し、結びつけることで、ユーザの状況に応じた種々のアクションを能動的に実行するプロアクティブ環境フレームワークを提案する。本フレームワークでは、ウェブ・コンテンツ間の連携動作を定義した ECA ルールの集合によって環境を構築する。また、連携定義はユーザによる GUI 上の対話的操作で定義可能である。提案フレームワークを用いることで、柔軟で容易なプロアクティブ環境が構築可能となる。

Proactive Environment Framework based on Meme Media Technology

MOTOYA NAKAYAMA,[†] HAJIME IMURA[†] and YUZURU TANAKA[†]

Recently, progress of the sensor network technology and the web technology have enabled users to easily get status of computer devices through the web. Users can demand a system for cooperating with the status information and an existing web application. In this paper, we propose a new framework "Proactive Environment Framework". This framework can execute various actions corresponding to users' situation "proactively". The framework uses ECA (Event-Condition-Action) rules for defining the cooperation among the web contents. In this framework, the environment is constructed with sets of ECA rules that define the coordinated operation among the web contents. This framework allows users to construct proactive environments flexibly and easily.

1. はじめに

近年、情報社会の発展に伴い環境中に存在する電子機器が増加した。それらがネットワークに接続されることで、ユーザは WWW(World Wide Web:以下ウェブ) を介して電子機器を操作することが可能となった。これらのウェブに接続されたリソースを環境要素と呼ぶ。これらから得られる情報や既存のウェブアプリケーションを連携させることで構築できる環境システムへの需要は高く、数々の研究が進められている。しかし、多くの場合これらは複雑なシステムを用いて制御されている。ユーザがネットワークを介して繋がっている異なる環境要素同士をアドホックに連携することは困難である。また環境中に存在するセンサー群などの数が膨大となると、それらから得られるウェブ情報を管理することが困難となる。ユーザがアドホックに環境要素を連携・管理するためには、新たなアプロー

チが必要となる。

環境要素を連携させ状況に対応したサービスを提供する技術の 1 つとしてウェアラブル・コンピューティングを応用したものがある。宮前ら⁵⁾ による研究ではウェアラブル・コンピューティング環境に対して、ユーザの状況に応じた柔軟なサービスを提供する基盤システムを提案している。ウェアラブル・コンピューティングとはユーザがセンサーなどのコンピュータを身体に装着して常に持ち運ぶシステムの形態である。コンピュータを常に携帯することで、ユーザの場所や状態に応じて地図情報などの必要な情報を提示するサービスを提供することが可能となる。ここでは XML によって記述された ECA ルールを用いてユーザの置かれた状況に応じた情報の提示を行うシステムを実現している。ECA ルールを追加することで、新たな状況に対応することができる。また、イベント・アクションのセットをプラグインの形式で提供し、プラグインを追加することで機能の拡張を実現している。

また、個々の環境要素が状況に応じて自律的に連携するシステム概念として Tenmenhouse により提唱

[†] 北海道大学大学院知識メディアラボラトリー
Meme Media Laboratory, Hokkaido University

されたプロアクティブ・コンピューティング²⁾がある。これは、コンピュータが状況を自律的に判断しユーザにとって最適な行動を予測しながら実行するというシステム概念である。現在ではユーザが数台のコンピュータを持ち、直接それらと情報のやり取りを行うことによりそれぞれのコンピュータを管理・操作している。しかし、今後コンピュータ・デバイスが増加した状況において、コンピュータは人間の要求を予測し人間に代わって要求を実行することが必要とされる。現在のインタラクティブ・コンピューティングとの違いは、コンピュータがユーザからの命令を待つのではなくコンピュータがユーザからの命令や要求を予想し、判断することで自律的に行動を起こすことである。このシステムを用いた応用例としては、センサー、コンピュータおよび通信を組み合わせたセンサー・ネットワーク³⁾が挙げられている。

本稿では環境要素を連携させ、自動的にユーザの置かれている状況に必要なアクションを実行する環境をプロアクティブ環境とする。また、ユーザが膨大なウェブ情報の中から必要な情報を抽出し管理する技術として知識メディア技術を用いる。さらに、環境要素間の連携定義をECAルールによって行う。

本論文ではまず、第2章においてフレームワーク実現のための基盤技術として用いる IntelligentPad とその上で実現されたウェブ情報を抽出する手法である C3W について述べる。また、抽出した各要素間の連携を定義するために用いる ECA ルールについても述べる。第3章では、プロアクティブ環境についてのフレームワークおよびアーキテクチャの詳細を述べる。第4章では、フレームワークの実現例としてセミナーの開催を例にして説明する。フレームワーク実現に必要な要素を連携する手順を述べ、セミナー支援アプリケーションの構築について述べる。

2. 要素技術

ウェブから情報を抽出し、それをシステム環境内でメディアオブジェクトとして扱うための技術として知識メディア技術¹⁾がある。本研究では知識メディア技術を実現する基盤技術である IntelligentPad と、さらに IntelligentPad 上で実現されたウェブ情報を抽出する技術である C3W をウェブからの情報の取得に用いる。また、抽出した各要素を結びつけ、連携動作を定義するためにはECAルールを用いる。これらについて以下に述べていく。

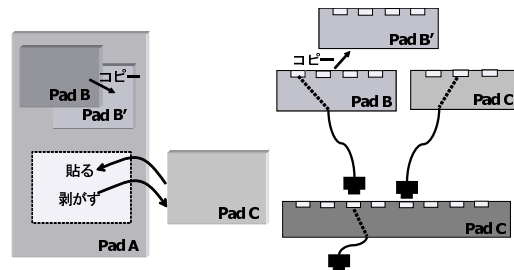


図1 パッドの貼り合わせとスロット結合

2.1 IntelligentPad

IntelligentPad システムとは計算機上で表すことのできるあらゆる情報をメディアオブジェクトとして統一的に扱うことができるシステムである。文字や数字、画像や音声などを計算機上で実現されるあらゆるアプリケーションなどはパッド (Pad) とよばれる可視化されたメディアオブジェクトにより表現される。

パッドは紙のメタファをもったメディアオブジェクトとして表現され、ユーザが直接的な操作により編集・再利用を行うことができる。様々な機能を持ったパッドを貼り合わせることで、それらの機能が編集し統合された一つの知的資源とすることができ、またそれらを剥がすことにより機能を分解することが可能である。また、同時に張り合わせたパッドの間で行われるデータの受け渡しにより連携を定義することができる。貼り合わされたパッド間の機能連携は各パッドが持つスロットを介して行われる。パッドは、内部状態をスロットを通してのみ外部に公開している。



図2 C3W の概要

2.2 C3W

ウェブから情報を抽出する手法として C3W(Clip, Connect, Clone for the Web)⁴⁾を用いる。C3W は IntelligentPad 上で実現されたシステムで、ウェブ上のコンポーネントをその機能を保持したまま抜き出し、

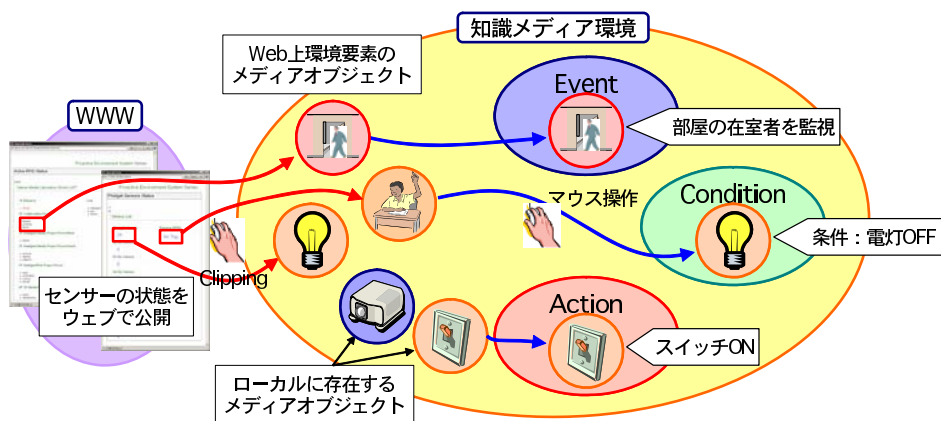


図 3 フレームワークの概要

IntelligentPad 環境内のパッドとして扱うことができる。この IntelligentPad 環境内に Clipping したウェブ上のオブジェクト **ViewPort** とよび、そのパッド表現を **ViewPort** パッドと呼ぶ。

本研究では C3W の Clipping の機能を用いてウェブページ中のコンポーネントをパッドとして抽出し利用している。

2.3 ECA ルール

本研究では環境要素の連携定義に ECA(Event-Condition-Action) ルールを用いる。ECA ルールとは、アクティブデータベース⁶⁾の分野で用いられている動作記述言語である。アクティブデータベースとは、データベース内部で事象が発生した際に、あらかじめ定義された条件に応じて、値の更新などの操作を自動的に行なうシステムである。ECA ルールは Event, Condition, Action の3つを1組にして記述する。各要素はそれぞれ、発生する事象 (Event), ルールの発火条件 (Condition), 実行される操作 (Action) を表現している。

3. プロアクティブ環境

本研究では環境要素を連携させ、自動的にユーザの置かれている状況に必要なアクションを実行する環境をプロアクティブ環境とする。そのために、ウェブから環境要素の状態を抽出し、さらに抽出したオブジェクトを ECA ルールによって連携定義する。ここで、各オブジェクトは Event, Condition, Action の流れに沿った評価がなされる。これらの ECA ルールの集合により、様々な状況に応じたアクションを実行するプロアクティブ環境を実現する。

以下ではプロアクティブ環境を実現するフレーム

ワークおよびアーキテクチャについて説明する。

3.1 フレームワーク

提案フレームワークでは、ウェブ上に存在する環境要素をユーザが容易に組み替え、連携定義することを実現する。本フレームワークを用いることで、ユーザは専門的な知識を必要としない GUI 上の操作のみで環境要素を連携することが可能となる。

図 3 はフレームワークの概要を表している。ユーザはマウス操作でウェブから情報を抽出する。ここで抽出された情報箇所は GUI 操作可能なパッドとして表現される。これらを ECA ルールに沿って Event, Condition, Action の3つの要素に貼り合わせ連携定義する。

図 3 の例では、ユーザはマウス操作によってセンサー情報が公開されたウェブページから部屋の状態を Clipping する。そして生成されたパッドを Event 部に貼り合わせる。さらに部屋の照度の状態を Clipping したパッドを Condition 部に貼り合わせ、条件である‘電灯 OFF’を設定する。そして Action 部にはメール送信機能を持つパッドを貼り合わせる。以上の操作により、部屋の変化を検知し、条件を評価して、成立した場合にユーザが入室時に自動的に電気をつけるアプリケーションを構築することが可能となる。

本フレームワークではユーザの GUI による対話的操作によって ECA ルールを生成することができる。これによりユーザはルール記述やプログラミング能力などの知識を持たずに、単純な GUI 操作のみで複雑な連携アプリケーションを構築することが可能となる。

3.2 アーキテクチャ

提案フレームワークは環境要素を ECA ルールにもとづいて連携することで実現する。図 4 はアーキテク

チャの内部構造を表したものである。ここでは4つのコンポーネントを用いる。環境要素の変化 (Event) を検知する EventDetection パッド、条件 (Condition) を評価する Condition パッド、あらかじめ定義されたアクション (Action) を実行する Action パッド、そしてこの3つのコンポーネントをECAルールにもとづいて評価する RuleEval パッドである。

ユーザがウェブから情報を抽出する場合、C3WのClippingの機能によってメディアオブジェクトとしてパッド化される。RuleEvalパッドではパッドの状態の変化をEventDetectionパッドを通じて検知する。Conditionパッドも同様にClippingされたメディアオブジェクトの状態の変化を条件に沿って評価し、結果をRuleEvalパッドに伝搬する。その結果、RuleEvalパッドはActionパッドへトリガーを送信し、あらかじめ定義されたアクションを実行する。

ここで本アーキテクチャの中心となる、ルール評価における値の伝搬とルール評価の制御手法の2点について詳細に述べる。

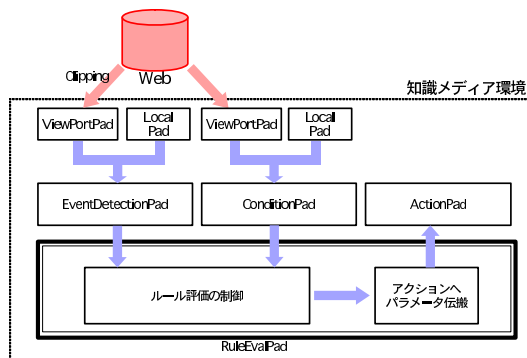


図4 アーキテクチャの概要図

3.2.1 値の伝搬

各コンポーネント間の値の伝搬による協調動作は、RuleEvalパッドを通じて行われる。

EventDetectionパッド群

EventDetectionパッドでは、パッドの状態をスロットを介して取得する。そして、取得した情報が変化した場合にRuleEvalパッドへトリガーを送信する。ここでは変化した状態のデータを送信するのではなく、ブール型 (True, False) のデータで送信する。

Conditionパッド群

Conditionパッドでは、EventDetectionパッドと同様に対象となるパッドの状態をスロットを介して取

得する。さらにConditionパッドでは取得した状態をあらかじめ設定された条件をもとに評価を行う。図5にConditionパッド群の構造を示す。Conditionパッド群の機能は以下に示したパッドを組み合わせて実現する。

○ ConditionEvalパッド

ConditionEvalパッドでは条件にもとづいてViewPortパッドの状態の評価を行う。このパッドは評価に用いる値 (Parameter) とその演算子 (Operator)、さらに貼り合わされているパッドの状態を下段のパッドに伝播する際に付加するキー (KeyName) をスロットの値として保持している。その条件をもとに評価した結果をスロットを介して下段のパッドに伝搬する。この内容はブール型の値と、KeyNameを付加した上段のViewPortパッドのスロットの値 [True/False, < KeyName, Value >] である。ここで設定するKeyNameはConditionパッド、RuleEvalパッドを介してActionパッドに伝搬される。その際にActionパッドの実行時のパラメータスロット名と一致するものを設定する必要がある。

○ ANDパッド・ORパッド

ANDパッド・ORパッドは複数の条件を組み合わせる場合に、上段の2つのパッドから伝搬された結果に対して論理演算を行う。伝搬された結果の先頭に含まれるブール型の値に対してAND・ORの演算により新たなブール型の値を取得する。下段のパッドには新たなブール型の値と上段より伝搬された < KeyName, Value > のリストを連結して伝搬する。

○ Conditionパッド

Conditionパッドは上段より伝搬された結果 [True/False, < KeyName1, Value1 >, ..., < KeyNameN, ValueN >] をRuleEvalパッドに伝搬する。

以上の機能を貼り合わせることでConditionパッド群を構成する。その手順は、まずViewPortパッドをConditionEvalパッドに貼り合わせる。その際にParameter, Operatorの各値を設定する。またその値がActionパッドの実行時のパラメータとして必要な場合はKeyNameにActionパッドのパラメータスロット名を設定する。複数の条件を組み合わせる場合はANDパッド・ORパッドを適当な位置に挟むことで対応する。Conditionパッドの上にごここまで組み合わせたパッド群を貼り合わせることでConditionパッド群を構成する。

Action パッド群

Action パッドは RuleEval パッドからのトリガーを受けてあらかじめ設定されたアクションを実行する。前段で Condition パッドから条件が成立した結果を受けた RuleEval パッドは、Action パッドへアクション実行のトリガーを送信する。このトリガーの内容は全段の Condition パッド群の結果から先頭のブール型の値を取り除いた、 $\langle \text{KeyName}, \text{Value} \rangle$ のリストである。RuleEval パッドは Condition パッドからの結果が *False* であるならば *Null* データを、*True* であるならば前述のパラメータのリスト型データを伝搬する。Action パッドではトリガーの発行の判別を値が *Null* であるか、パラメータのリストであるかによって判別する。Action パッドは自身の保有するパラメータスロット名とパラメータリストの *KeyName* を比較し、適当な値を引数としてあらかじめ定義されたアクションを実行する。

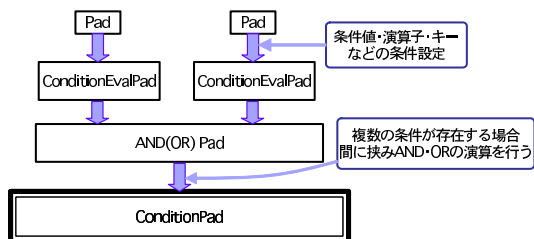


図 5 Condition パッド群の構造

3.2.2 ルール評価の制御手法

各 ViewPort パッドはウェブからの情報の取得のため、一定間隔のポーリングを行っている。しかし、あるイベントが発生した際に情報の更新間隔によってはリアルタイムなコンディションの評価が困難なケースも想定される。また、ウェブからの情報自体も更新間隔が存在し、一定間隔によってしか情報が更新されない。このような例は「地震発生の速報」とそれによる「津波警報」などの場合に見られる問題である。このような例では、「地震速報」から「津波の警報」までは一般に数分のタイムラグが生じるだろう。これらの問題点に対して本アーキテクチャでは ECA ルールを非同期に評価することで解決する。非同期の ECA ルールの制御については図 6 に示すルール制御手法を用いる。

ユーザは EventDetecton パッドを RuleEval パッドに貼り合わせる際にイベント有効期限 ($VLimit$) を RuleEval パッドに設定する。これによりあるイベン

トが検知された際に、イベントが検知された時点からイベント有効期限まで各条件 (ConditionEval パッド) の状態を監視し、これを評価する。その結果をコンディション成立条件として用いる。また、複数回 *True* が検知された場合に、これらをまとめて 1 回の *True* の結果が生じたと見なすか、あるいはそのまま複数回の *True* の結果と見なすかはユーザが RuleEval パッドの *MultiOn* スロットの値として設定可能となっている。これにより前述のような、イベント発生時に情報更新のタイムラグのためリアルタイムのルール評価をすることが困難な問題を解決することができる。

また、ルール成立時にアクションを実行する際にも制御が必要となる場合が考えられる。1 度だけ実行したいアクションと複数回実行したいアクションの 2 つがあり得るであろうこのため、ルールに設定された条件が成立している間はアクションを複数回実行するかどうかを制御する必要がある。本フレームワークでは、ユーザが Action パッドに対して連続実行を Action パッドの *MultiOn* スロットによって設定可能となっている。これにより、実行するアクションに応じた設定が可能となる。

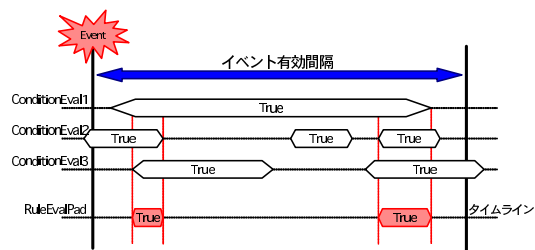


図 6 ルール制御

4. 実現例：セミナー支援システム

提案のプロアクティブ環境フレームワークを用いた実現例として、セミナーの開催を用いて解説する。ここで、セミナーを開催する部屋には、以下のセンサー、アクチュエータ群を配置してあるものとする。

- Phidgets Inc. 社⁸⁾ のセンサー群
 - 照度センサー
 - パッシブ型 RFID タグリーダー
 - サーボモータ
- RF Code 社⁹⁾ アクティブ型 RFID タグリーダー
 - さらに各センサー・アクティブ型 RFID タグリーダーが検出した状態はウェブページ (CGI アプリケーション) として公開する。これによってウェブを介して各センサーの状態を取得することができるようになる。



図 7 センサー・アクチュエータ群の配置

また、サーボモータについてもウェブ上の CGI アプリケーションを用いて操作できるようにした。

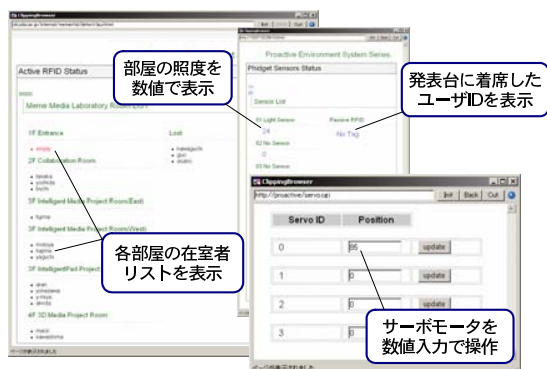


図 8 センサー群の状態をウェブ上で公開

これらを踏まえて、以下の実現例を解説していく (図 9 参照)。

- (1) セミナーが開催される部屋の在室人数をアクティブ型 RFID タグによって取得
- (2) 参加者全員が揃ったことを検出
- (3) 部屋の電灯を消灯

まず、各部屋の滞在者リストが表示されるウェブページからセミナーの開催される部屋の部分を Clipping し ViewPort パッドを得る。次にこれをコピーし Event 部、および Condition 部に貼り合わせて、Condition 部の *Parameter* にセミナー参加人数の '8'、*Operator* に演算子 '<=' を設定する。そして、Action 部には部屋の照明のスイッチを操作する Action パッドを貼り合わせる。これによってセミナー時に参加者が全員揃うと、自動的に照明を調整するシステムが構築できる。この例での ECA ルールは以下に示す通りである。

Event: 部屋の在室人数の変化

Condition: 在室人数が 8 人以上 (参加者数)

Action: 部屋の照明を消灯

これに新たな ECA コンポーネントを追加すること

で機能を拡張する。ここで、スクリーン横に置かれた台に内蔵されたパッシブ・RFID タグリーダが設置されているものとする。実現する機能は以下の通りである。

- (1) パッシブ型 RFID タグリーダの状態 (*UserID*) を取得
- (2) 部屋の照度が閾値以下になったことを検知
- (3) *UserID* に応じたプレゼンテーション用のスライドを投影

この ECA コンポーネントの構成の手順はまず、パッシブ・RFID タグリーダの状態をウェブから Clipping し ViewPort パッドを得る。次にこれをコピーして、Event 部および Condition 部に貼り合わせる。ただし、ここでは複数の Condition を設定する必要がある。そのためにまず 1 つめの条件として、パッシブ・RFID タグリーダの ViewPort パッドを Condition 部に貼り合わせ *Parameter* にタグが存在しない状態の 'No Tag'、*Operator* に演算子 '! ='、*KeyName* に '*UserID*' を設定する。さらに 2 つめの条件として、部屋の照度を取得する照度センサーの状態をウェブから Clipping し、Condition 部に貼り合わせ、*Parameter* に照明 OFF 値の '20'、*Operator* に演算子 '<=' を設定する。これらの 2 つの Condition コンポーネント間には AND パッドを挟むことによって AND 演算を設定する。Action 部にはプロジェクトを通じてプレゼンテーションを実行する Action パッドを貼り合わせる。Action 部では、自身のスロット名と Condition 部の結果の *KeyName* を比較し、発表するユーザの *UserID* を取得し、*UserID* に対応するファイルを実行する。これらによりユーザが発表位置に着いたときに部屋の状態を確認し、それぞれのユーザに対応したプレゼンテーションを開始するルールを追加することができる。また、この例での ECA ルールは以下に示す通りである。

Event: パッシブ型 RFID タグリーダの状態の変化
Condition₁: *UserID* が 'No Tag' 以外

Condition₂: 部屋の照度が規定値以下

Action: 発表者に対応したスライドの開始

以上に述べた ECA コンポーネントを構築し機能拡張することでセミナー開始時に部屋の調光、発表者に応じたプロジェクトの制御を行うシステムが実現できる。これによりユーザは高度なプログラミングの知識を持たずに、複雑なセミナー支援アプリケーションを作成することが可能となる。また、各 ECA 部品はそれぞれ交換可能でスライドの投影の代わりにデモンストレーション・ムービーの再生を行う、といった変更

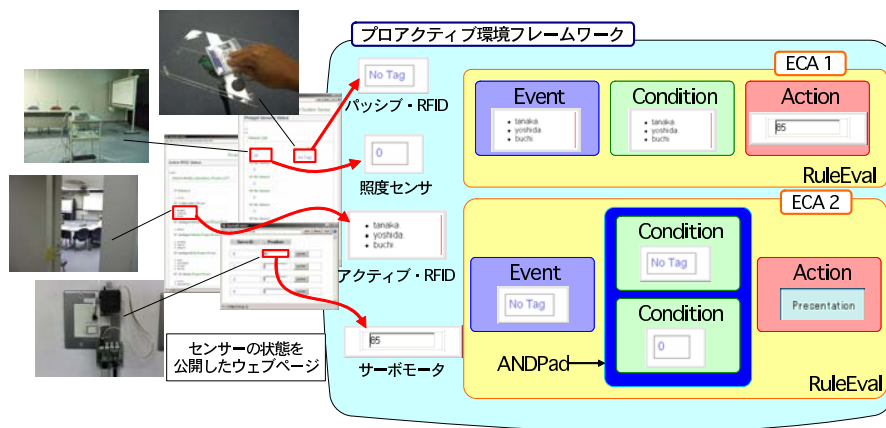


図9 実現例の概念図

も容易に可能である。

5. おわりに

本稿ではプロアクティブ環境を実現するフレームワークについて提案し、IntelligentPad上にこれを構築した。このシステムでは、ウェブ上に公開された環境情報やウェブアプリケーションなどをECAルールによって結びつけ、連携動作を定義することによってプロアクティブ環境を実現する。また、ECAルールを追加することで様々な状況に応じた動作を柔軟に拡張可能である。これによって、ウェブアプリケーションやシステムの開発者でなくとも、ウェブのエンドユーザがGUI上の対話的操作によって容易にプロアクティブ環境を構築可能となる。今後の展開としては、アクションをウェブ上のリソースに対して作用させ、ウェブを介して複数のシステムが連携することで、より大規模なシステムの構築が可能となると考えられる。その場合には、ルールの整合制のチェック機構と、ECAルールがループすることによる異常動作の検出が必要となると考えられるため、これらについての研究を進めていく予定である。

参考文献

- 1) Yuzuru Tanaka, Meme Media and Meme Market Architectures: Knowledge Media for Editing, Distributing, and Managing Intellectual Resources. July 2003, Wiley-IEEE Press.
- 2) David Tennenhouse, PROACTIVE COMPUTING, Communications of the ACM, Volume 43 Issue 5, May 2000.
- 3) J. M. Kahn, R. H. Katz, and K. S. Pister, Next century challenges: mobile networking for "Smart Dust". In Proceedings of the 5th Annual

ACM/IEEE international Conference on Mobile Computing and Networking. MobiCom '99. ACM Press, New York, NY, pp.271-278, 1999.

- 4) Jun Fujima, Aran Lunzer, Kasper Hornbak, Yuzuru Tanaka. Clip, Connect, Clone: Combining Application Elements to Build Custom Interfaces for Information Access, In Proceedings of the 17th Annual ACM Symposium on User interface Software and Technology, UIST '04. ACM Press pp.175-184, 2004.
- 5) 宮前雅一, 寺田 努, 岸野泰恵, 塚本昌彦, 西尾章治郎, ウェアラブル環境のためのイベント駆動型ナビゲーションプラットフォーム, 情報処理学会論文誌, Vol.46, No.3, pp.694-703, 2005.
- 6) Norman W. Paton, Oscar Díaz, Active database systems, ACM Comput. Surv. 31, 1 (Mar. 1999), 63-103, 1999.
- 7) RFIDハンドブック 第2版, Klaus Finkenzeller: ソフト工学研究所 訳, 日刊工業新聞社, 2004.
- 8) Phidgets Inc., <http://www.phidgets.com/>
- 9) RF CODE, <http://www.rfcode.com/>