

# ECAルールによる知識メディアの動的オーケストレーション

三浦 陽介      田中 譲

北海道大学大学院 情報科学研究科

コンピュータサイエンス専攻 知識メディア研究室

## 1 はじめに

Web 上には多種多様な情報源が存在し、常に最新の情報が発信されている。ニュースや株式相場などのような、刻一刻と更新されるコンテンツがその代表的な例として挙げられる。しかし、これらの情報源は相互連携を意図して構築されているわけではなく、独立のものである。よって、エンドユーザがその情報の鮮度を活かすには、多くの労力が必要であり、効率的に情報を利用するのは難しい。例えば、ニュースと株価の関係をリアルタイムに知ろうとした場合、ニュースサイトと株式相場のサイトの両方からの情報収集を行わなければならない。多くの情報にアクセスできるようになった反面、その中からユーザにとって必要な情報をいかに抽出するのが課題となる。

こうした現状をふまえて、本研究ではその解決のために、筆者らが研究開発する IntelligentPad[1] を基盤とし、情報の鮮度を活かし効率的に利用するためのフレームワークを提案する。また、本研究ではこのフレームワークのことをオーケストレーションと呼ぶ。IntelligentPad とは、Web 上の様々な資源を知識メディアオブジェクトとして部品化し、ユーザが画面上で自由に扱うことができるシステムである。本研究では、それらに対して ECA ルール (Event-Condition-Action rules) を適用し、情報獲得の自動化を図ることによってフレームワークを実現する。具体的には、情報の更新をイベントとして検知し、それを契機にルールに従って情報の評価を行い、特定の行動を起こすシステムを構築する。

## 2 知識メディア IntelligentPad

IntelligentPad は、計算機上に存在するあらゆる知的資源をパッド (Pad) と呼ばれる可視化された部品として外在化することにより、これらを統一的に扱い、知的資源をパッドとして編集することが可能になる。

これにより、複合文書 (Compound Document) の編集・流通を行うことを可能とするシステムである。各パッドは状態データを保持する任意個のスロットと呼ばれる内部オブジェクトを持ち、パッド間の情報の受け渡しは、そのスロットに対して、set, gimme, update の3種類のメッセージを送信することによってなされる (図1)。

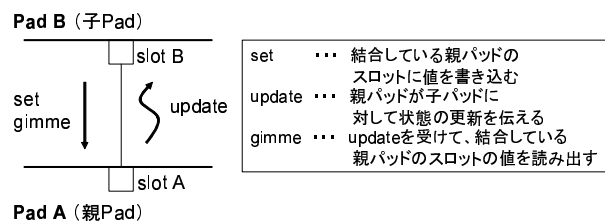


図1: IntelligentPad のパッド間標準メッセージ

本研究では、IntelligentPad の一実装である CHIP[2] を利用する。CHIP とはネットワーク通信機能を持った GUI 部品をホストコンピュータ上で動かし、それらの部品を画面上で貼り合わせることでリアルタイムに分散アプリケーションを構築する事が可能なシステムである。Web コンテンツをラッピングした CHIP は任意の HTML ノードをスロット化可能であり、Web アプリケーションの入力フォームや出力結果等のノードをスロット化し、別の Web コンテンツをラッピングした CHIP と貼り合わせることで、複数の Web アプリケーションを連携させたシステムを構築可能となる。

## 3 ECAルール

ECAルール (Event-Condition-Action rules) とは、発生する事象 (イベント)、ルールの発火条件 (コンディション)、実行される操作 (アクション) の3つを1組として記述するものである。情報の鮮度という観点から見て、イベントとなるべきは情報の更新である。IntelligentPad において、情報の更新は Pad 間に情報の伝達、つまりパッド間でスロットを介してメッセージの受け渡し (set, gimme, update) が発生したということと同義である。そこで、本研

Dynamic orchestration of Knowledge media  
by ECA rules  
Yosuke Miura,  
Yuzuru Tanaka  
Meme Media Laboratory, Hokkaido University  
N13W8, kita-ku, Sapporo, 060 8628, Japan

究では、これら Pad 間での情報の伝達をイベントとしてとらえる。そして伝達が起こった時の特定の Pad の値をコンディションとしてルールと比較・評価し、アクションを起こすシステムを構築する。これにより、情報の更新を即座に検知し、情報を得ることが可能となる。本研究では ECA ルールの実装に Prolog 言語を用いた。

#### 4 オークストレーションのフレームワーク

オークストレーションを実現するにあたって、EventDetectorPad と AgentPad を開発した。EventDetectorPad は、パッド間の情報の伝達を検知し、AgentPad へトリガーをかける機能を持ったパッドである。CHIP は各パッドごとに自身の URL を持っているのので、スロット結合を介さずに情報を伝達することが可能である。AgentPad は、RuleServer にルールの評価をさせ、その結果を受けアクションを起こす機能を持ったパッドである。イベントを検知したいパッドとパッドの間に、EventDetectorPad を挿入することによって情報の伝達を検知する(図2)。EventDetectorPad は、伝達された情報を拒否することなく下のパッドへ透過するので、それまでの機能を損なうことはなく、見かけ上、パッド間の構造には何の変化も無いものとして扱う事ができる。また、検知したい箇所が一部分とは限らない。EventDetectorPad は、様々な場所に仕掛けておくことができる。

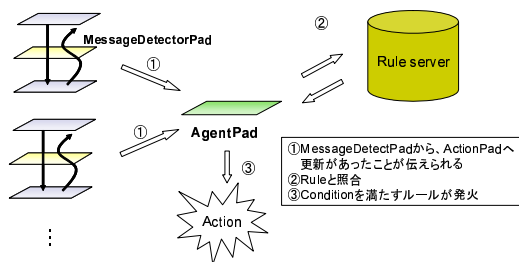


図 2: オークストレーションのフレームワーク

#### 5 応用例

応用例として、株価の Web ページを監視し、ルールで指定した条件に合致した時に証券会社の Web ページ上から注文を出すというシステムを示す。まず、株価の Web ページと、証券会社の Web ページを CHIP を用いてラッピングしておく。次に、目的の銘柄の株価の部分のスロット化し、間に EventDetectorPad を挿入する。株価が更新されるたびに、EventDetectorPad がイベントとして検知し、AgentPad に知らせる。それを受けて、AgentPad と RuleServer は目的

の株価を取得し、合致するルールがあるかどうか評価する。もし設定した値に達していれば、ルールに記述されているアクション等(売り注文, 買い注文)をあらかじめラッピングした証券会社の Web ページに対して行う。

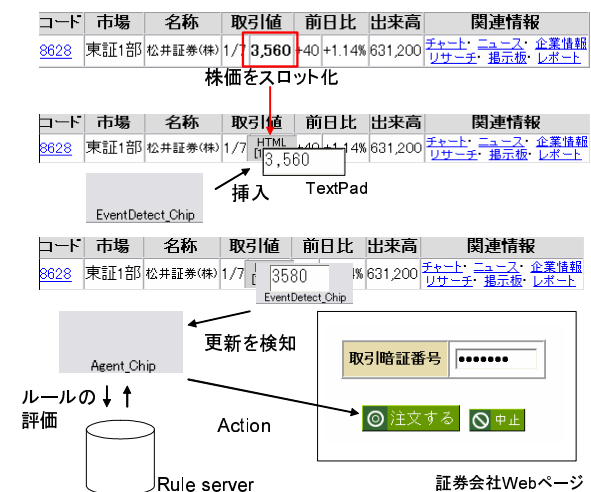


図 3: 応用例

#### 6 おわりに

本研究では、知識メディア IntelligentPad に ECA ルールを用いることによって、エンドユーザが情報の鮮度を落とさずに効率的に利用するためのフレームワークを提案した。

これにより、各情報源からの情報を即座に、そして統合的に利用することが可能となる。今後は、イベントの範囲を拡大し、ユーザの行動等を含めることによって、本フレームワークの有用性を更に高めていく。

#### 参考文献

- [1] Y.Tanaka: *Meme media and meme market architectures* Knowledge Media for Editing, Distributing, and Managing Intellectual Resources (2003)
- [2] Kimihito Ito and Yuzuru Tanaka: A Visual Environment for Web Application Composition, Proceedings of the fourteenth ACM conference on Hypertext and hypermedia, (August 26-30, 2003, Nottingham, United Kingdom). pp.184 - 193.
- [3] CHIP(Corroborating Host-Independent Pads), <http://km.meme.hokudai.ac.jp/people/itok/CHIP/index.J.html>