Vol.2013-DPS-156 No.7
Vol.2013-GN-89 No.7
Vol.2013-EIP-61 No.7
2013/9/12

Reteアルゴリズムを用いた家庭内エネルギー管理システムの複数スマートタップによる実装

川上 朋也[1]　義久 智樹[2]　藤田 直生[1]　塚本 昌彦[1]

概要：近年，センサ機器の普及により家庭内でも状況情報を容易に取得でき，状況に応じた家庭内エネルギー管理システム（Home Energy Management System; HEMS）への期待が高まっている．我々の研究グループでは現在，ルールに基づく HEMS において，処理の効率化のために Rete アルゴリズムを用いた手法を検討している．また，検討手法では，多数かつ密な電化製品やセンサの設置に対応するため，電化製品やセンサの情報をネットワーク上のスマートタップと呼ばれる機器に分散させる．本稿では，Rete アルゴリズムを用いた HEMS の複数スマートタップによる実装について述べる．

キーワード：HEMS，分散処理，センサデータ，ルールベースシステム，Rete アルゴリズム

An Implementation of Home Energy Management System Using the Rete Algorithm on Smart Taps

Abstract: In recent years, information about a situation are easy to collect in home because sensors become popular and varioius sensors are placed. In this environment, Home Enegry Management System (HEMS) which controls electrical appliances based on those situation is largely expected. Currently, we have proposed a rule-based HEMS using the Rete algorithm. In the proposed system, information about electrical appliances and sensor data are distributed to "smart taps" in network to deal with a huge amount of electrical appliances and sensors. In this paper, we explain an implementation of the proposed HEMS using the Rete algorithm on smart taps.

Keywords: HEMS, distributed processing, sensor data, rule-based system, Rete algorithm

1. はじめに

近年，スマートフォンなどの携帯端末を含め，センシング機能をもつさまざまな機器が普及している．これにより，家庭内でも状況を把握するための情報を容易に収集することができ，いわゆる「スマートホーム」を実現できる．スマートホームにおいては，センサデータなどから推定された状況に基づき，QoL（Quality of Life）を下げずに電化製品を制御することで節電を行う家庭内エネルギー管理システム（Home Energy Management System; HEMS）が期待される．冗長なエネルギー消費を抑えるために各家庭でも電力消費の抑制が求められ，電力要求が許可されてから管理システム側が給電する EoD（Energy On Demand）をはじめ，HEMS は特に多くの注目を集めている [1,2]．

現在，電力制御を行うルールベースの HEMS が数多く提案されている．これら既存手法の多くは IF-THEN 形式のルールを想定しており，我々の研究グループでも関連する手法を提案している [3–5]．IF-THEN 形式のルールはプロダクションシステム [6] によって扱うことができ，代表的な照合アルゴリズムに Rete アルゴリズムがある [7]．また，Rete アルゴリズムの処理を複数のコンピュータで行う手法も提案されている [8]．将来的には家庭内にさらに多数かつ密な電化製品やセンサの設置が予想され，この分散処理モデルは可用性やスケーラビリティの点で有効であると考えられる．しかし，既存手法は Rete アルゴリズムの一部の処理を分散させることを目的としており，特定のコ

[1] 神戸大学大学院工学研究科
Graduate School of Engineering, Kobe University
[2] 大阪大学サイバーメディアセンター
Cybermedia Center, Osaka University

Vol.2013-DPS-156 No.7
Vol.2013-GN-89 No.7
Vol.2013-EIP-61 No.7
2013/9/12

ンピュータに負荷が集中する可能性がある．

我々の研究グループでは，Rete アルゴリズムを用いたルールベースの HEMS を検討している．検討システムでは，電化製品やセンサの情報をネットワーク上の「スマートタップ [9]」と呼ばれる機器に分散させる．スマートタップはコンセントと電化製品の間に接続し，消費電力の測定や供給電力の制御，特定のコンピュータプログラムの実行などを行う．検討システムではスマートタップが状況に応じて電化製品への電力を制御し，状況の把握はルールに基づいて行う．そのため，ルール処理およびデータ収集による負荷は，各スマートタップで分散される．また，Rete アルゴリズムに基づいてルールを処理することで，処理およびデータ収集の回数を削減する．本研究では，検討システムの複数のスマートタップによる実装について述べる．

以下，2 章で Rete アルゴリズムについて述べる．3 章で Rete アルゴリズムを用いたルールベースの HEMS を説明し，4 章で複数のスマートタップを用いた実装について述べる．最後に 5 章で本稿をまとめる．

2. Rete アルゴリズム

IF-THEN 形式のルールに基づいて推論するプロダクションシステムは，ルールを記憶するプロダクションメモリ（PM），ワーキングメモリエレメント（WME）と呼ばれるデータを記憶するワーキングメモリ（WM），推論エンジンと呼ばれる制御プログラムから構成される．

プロダクションシステムでは条件照合の高速化に関するさまざまな研究が行われており，Forgy によって提案された Rete アルゴリズムは代表的な条件照合アルゴリズムである [7]．Rete アルゴリズムは，Rete ネットワークと呼ばれるグラフ構造をルールに基づいて作成する．Rete ネットワークでは，入力データをルールの条件によってフィルタリングする節（ノード）を α ノード，β ノードと呼ぶ．α ノードは，1 つの親からの入力データを条件によってフィルタリングする．一方，β ノードは，2 つの親からの入力データを結合し，結合データを条件によってフィルタリングする．この β ノードにおける処理は “join” と呼ばれる．これら α ノード，β ノードで条件を満たしたデータは下流のノードへ送られる．Rete アルゴリズムは α ノードや β ノードでの条件照合において，条件を満たすデータあるいはデータの組合せを，それぞれ α メモリ，β メモリとして記憶しておき，データが追加や削除された場合に必要となる条件照合処理を削減している．Rete アルゴリズムは Drools[*1] や JESS[*2] など，多くのプロダクションシステム構築ツールやルールエンジンで用いられている．また，データ更新が頻繁に発生する環境を想定し，α メモリや β メモリを保持しない TREAT アルゴリズム [10] や，条件照合の計算コストが高いルールに対処するための手法 [11] が提案されている．さらに，条件照合を高速化するためのワーキングメモリのデータ構造も提案されている [12]．

[*1] http://www.jboss.org/drools/
[*2] http://www.jessrules.com/

```
for electrical_appliance_in_entrance e,
    sensor_data_in_living_room s
if e.type = “light” and
    e.status = “OFF” and
    s.type = “human detection” and
    s.value = TRUE
then
    turn e on
```

図 1 擬似コードによるルールの例
Fig. 1 An example of rule written in pseudocode

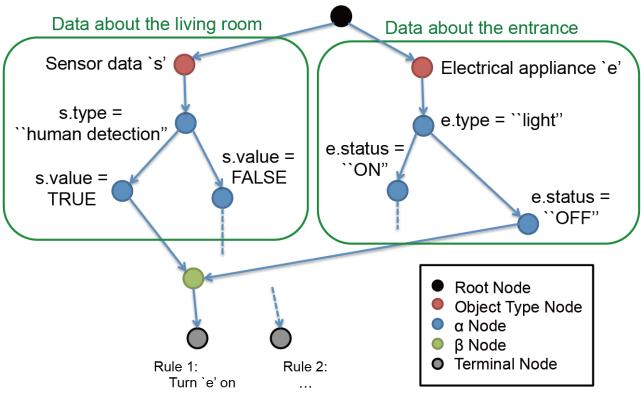

図 2 Rete ネットワークの例
Fig. 2 An example of Rete network

Rete アルゴリズムについて，**図 1** に示したルール例を用いて説明する．図 1 は「居間に人が存在すれば，玄関の電灯を点ける」というルールである．図 1 のルールから生成される Rete ネットワークを**図 2** に示す．ルールを満たすコンテンツの抽出は，図 2 の最上流に “Root Node” で示したノードから電化製品やセンサのデータを流すことで行われる．データはまず種類に応じて，下流の Object Type ノードと呼ばれる “sensor data s”，“electrical appliance e” へと振り分けて流される．同様に各ノードで示した条件を満たすデータのみをさらに下流のノードへ流すことで，最終的にルールを満たすデータの組みが図 2 の最下流に “Terminal Nodes” で示したノードに流される．Terminal Node はルールごとに存在し，ルールを満たすデータは対応する Terminal Node を確認することで取得できる．図 2 の Terminal Node では，左側に「居間に人が存在し，点いていない玄関の電灯」のデータが流される．このとき，α ノード，β ノードには，処理結果がそれぞれ記憶されている．例えば，「居間に人が存在しない」という条件を含むルールを新たに追加する場合，“s.value = FALSE” で示した α ノードの下流にノードを追加することでこの α ノー

Vol.2013-DPS-156 No.7
Vol.2013-GN-89 No.7
Vol.2013-EIP-61 No.7
2013/9/12

ドがもつ α メモリの内容を取得でき，共通する条件に対する処理を抑制できる．

Rete アルゴリズムを分散環境の Publish/Subscribe (Pub/Sub) システム [13] へ適用した手法として，DHT (Distributed Hash Table) の一種である Chord [14] を用いた手法が提案されている [8]．文献 [8] では WME の主語，述語，目的語をそれぞれ Key とし，購読者のリストを Value として Chord で分散管理する．すなわち，1 つの WME につき，Key と Value の組を Chord に 3 つ登録する．また，多用される Key を担当するピアの負荷を分散するため，WME が有効となる時間的な情報（期間）を Key に付加することを議論している．しかし，すべての WME が時間的な情報を持つとは限らず，時間的な情報の付与および Chord 上での探索について具体的な手法を示していない．また，文献 [8] は Pub/Sub システムにおいて α メモリを共有することを目的としている．そのため，Rete アルゴリズムにおける join 処理は複数の α メモリを購読するコンピュータ自身が行い，処理結果の β メモリは共有されない．つまり，共通の join 処理が含まれるルールを複数コンピュータが用いる環境では，処理に必要なデータを各コンピュータが個別に収集する．収集するデータには join 処理により無効と判断されるデータも含まれ，冗長な通信が発生することになる．

3. Rete アルゴリズムを用いた家庭内エネルギー管理システム

本章では，我々が検討している Rete アルゴリズムを用いた HEMS および処理の割り当て方式について述べる．

3.1 システムの概要

検討システムでは電力制御を行うルールはネットワーク上のスマートタップ [9] で処理し，ルール処理および必要なデータを収集するための負荷を各スマートタップで分散させる．また，Rete アルゴリズムに基づいてルールを処理することで，処理やデータ収集の回数を削減する．検討システムでは，複数の処理結果を結合する β ノードの処理結果もスマートタップ間で共有する．

検討システムのモデルを**図 3** に示す．検討システムでは N_1, N_2, N_3 で示すようなスマートタップを家庭内に設置し，スマートタップはネットワーク経由で相互に情報の送受信を行う．電化製品は近隣や同じ部屋のスマートタップに接続し，スマートタップは接続された電化製品に対して消費電力などの測定や給電，制御を行う．状況を把握するために用いるセンサ機器はネットワークやスマートタップに接続し，観測データを他の機器へ送信する．スマートタップは電化製品やセンサ機器から情報を収集し，Rete アルゴリズムに基づいて IF-THEN ルールを処理する．ルールの処理結果もスマートタップが記憶し，Pub/Sub モデル

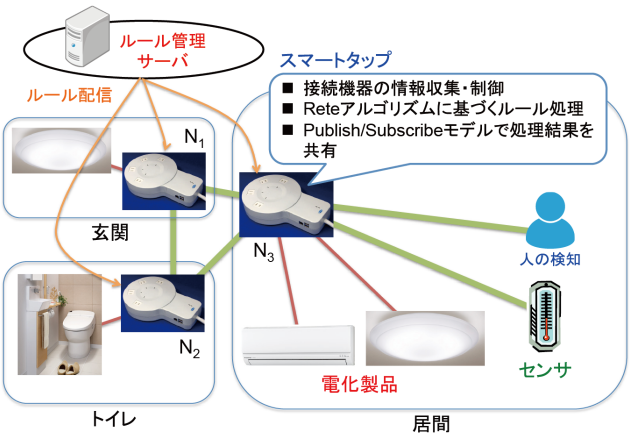

図 3 システムモデル
Fig. 3 System model

で共有する．ルールについては，ネットワーク上の特定のコンピュータが統合的に編集するものとし，更新があった場合にはスマートタップにルールを配信することで，すべてのスマートタップ間でルールの整合性を維持する．

3.2 スマートタップへの処理割り当て方式

本研究では，ルール処理および処理結果の記憶はルールの動作部に関係するスマートタップに割り当てる．

図 1, 図 2 の例におけるルール処理の流れを述べる．図 1, 図 2 の例では，図 1 のルールの動作部は玄関の電化製品に関連するため，玄関のスマートタップ N_1 がルールを処理する．まず，スマートタップ N_1 は接続された電化製品のうちで “type = light” かつ “status = OFF” の電灯が存在するかを確認し，ルール処理に必要なデータを居間のスマートタップから収集する．図 2 の例ではスマートタップ N_3 が居間にあり，N_1 は図 1 の “s.type = TRUE” に関連するデータの配信を N_3 に要求する．N_3 は要求されたデータを処理結果で確認し，該当するデータを N_1 へ配信する．要求されたデータを N_3 が保持していない場合，N_3 は関連するセンサ機器から情報を収集し，N_1 への送信および処理結果の記憶を行う．N_1 は “e.status = OFF” および N_3 から受信した “s.value = TRUE” に関係するデータを結合し，すべての条件を満たしていれば電灯 e を点ける．N_1 は処理結果を記憶し，状況に変更があった場合は，ルールで影響する箇所に反映することで最新の結果を維持する．

4. 複数スマートタップによるシステムの実装

4.1 スマートタップ

本研究では，3 章で述べたシステムを株式会社エネゲート[*3]が研究開発しているスマートタップを複数用いて実装した．実装システムに用いたスマートタップは Raspberry Pi Model-B[*4]をもとに構成されており，仕様を**表 1** に示す．

[*3] http://www.enegate.co.jp/
[*4] http://www.raspberrypi.org/

表 1 スマートタップの仕様
Table 1 Specifications of the smart tap

チップ	Broadcom BCM2835 SoC
CPU	ARM1176JZ-F 700MHz
GPU	Broadcom VideoCore IV
メモリ	512MB SDRAM
インタフェース	Wi-Fi，USB2.0，有線 LAN（RJ45）
OS	Debian GNU/Linux
コンセント数	4
測定項目	電力量，電力，電圧，電流
定格電圧	15V
定格電流	15A（コンセント 4 個口の合計）
制御定格	コンセント個別に最大 15A の負荷を ON/OFF

実装システムではスマートタップは Wi-Fi（IEEE 802.11n）経由でネットワークに接続し，無線 LAN アダプタには I-O DATA 社の WN-G150UMK を用いている．また，スマートタップはソフトウェアブレーカ機能をもち，さまざまな電力制御ポリシーに対して，コンセントごとの給電の開始および停止を柔軟に行える．スマートタップのハードウェアは主にアプリケーション基盤とセンシング・制御基盤で構成されており，各基盤は以下の機能をもつ．

- アプリケーション基盤
  - 外部とのデータ通信
  - センシング・制御基盤と通信することで，センサデータの取得やソフトウェアブレーカの制御指令を行う
  - 将来的なセンサ追加などに備えた USB2.0 インタフェース
- センシング・制御基盤
  - アプリケーション基盤に対してセンサデータを送信
  - アプリケーション基盤からの制御メッセージを受信し，ソフトウェアブレーカを制御

また，アプリケーション基盤で動作する Debian GNU/Linux には，以下の機能をもつソフトウェアが含まれている．

- 外部との通信
  - センサデータを外部に送信
  - 外部からの ON/OFF 制御メッセージや制御ポリシー設定を受信
- 制御ポリシーの管理・実行
  - センサデータと制御ポリシーに基づき，電力センサインタフェースがセンシング・制御基盤へ送信する制御指令の内容を決定
  - センサ・制御インタフェース
- センシング・制御インタフェース
  - センシング・制御基盤からのセンサデータを受信
  - センシング・制御基盤に対して制御指令を送信

実装システムでは，ルールに基づく家庭内エネルギー管理のために，スマートタップのコンセントに繋がっている電化製品の電力データを用いる．電力データは電流 [A]，電圧 [V]，電力 [W]，電力量 [Wh] で，スマートタップが各コンセントごとに 1 秒間隔で測定している．XML 形式による電力データの例を**図 4** に示す．また，前述のソフトウェアブレーカ機能は同様の XML 形式による制御コマンドをスマートタップに送信することで行う．コンセント番号 1 への給電を停止する制御コマンドの例を**図 5** に示す．

```
<root>
<info>
  <kind>command_socket</kind>
</info>
<data>
  <socket1><state>OFF</state></socket1>
</data>
</root>
```

図 5 制御コマンドの例
Fig. 5 An example of control command

表 2 ルールエンジンに登録される情報
Table 2 Information registered to rule enginge

電化製品	タップ ID タップの IP アドレス コンセント番号（1~4） 種類（テレビ，照明など） 状態（ON，OFF，省電力モードなど） 場所 優先度（1~10 など）
センサデータ	ID IP アドレス 種類（電圧，温度など） 状態 場所 測定値 誤差 測定時刻

4.2 ルールエンジン

実装システムでは，必要な情報の収集およびルール処理を各スマートタップが自律分散的に行う．スマートタップがルールを処理するためのルールエンジンは，Rete アルゴリズムに基づく処理が可能な Ruleby[*5] をもとに開発した．Ruleby は Ruby 形式で記述されたルールエンジンで，IF-THEN 形式のルールも同様に Ruby 形式で記述する．実装システムではスマートタップに Ruby 1.9.3 のインストールを行い，Ruleby を用いたルールエンジンを実行した．

ルール処理のためにルールエンジンに登録される情報を**表 2** に示す．登録される情報は，スマートタップに繋げら

[*5] https://github.com/codalytics/ruleby

Vol.2013-DPS-156 No.7
Vol.2013-GN-89 No.7
Vol.2013-EIP-61 No.7
2013/9/12

```
<root>
<info>
  <kind>notice_wattmeter</kind>
  <time>20130622080000008</time>
</info>
<data>
  <socket1><wh>52</wh><volt>100.913</volt><current>0.010</current><watt>0.0</watt><state>ON</state></socket1>
  <socket2><wh>0</wh><volt>100.972</volt><current>0.008</current><watt>0.0</watt><state>ON</state></socket2>
  <socket3><wh>0</wh><volt>100.924</volt><current>0.020</current><watt>0.0</watt><state>ON</state></socket3>
  <socket4><wh>12</wh><volt>100.951</volt><current>0.010</current><watt>0.0</watt><state>ON</state></socket4>
</data>
</root>
```

図 4 電力データの例
Fig. 4 An example of electric power data

```
rule :entrance_light,
[ElecAppliance, :ea,
  m.location == ENTRANCE_LOCATION,
  m.type == LIGHT_APPLIANCE_TYPE,
  m.status == OFF_STATUS],
[SensorData, :s,
  m.id == ID_FOR_TV_ON_LIVING_ROOM,
  m.type == WATT_SENSOR_TYPE,
  m.value > 1] do |context|
  set_status(context[:ea].ip,  # IP of the tap
    context[:ea].socket_id, "ON")
  context[:ea].status = ON_STATUS
  modify context[:ea]
end
```

図 6 Ruby 形式によるルールの例
Fig. 6 An example of rule written in the Ruby

れた電化製品の情報，および，スマートタップが測定する電力データを含めたセンサデータである．実装システムでは，それぞれ Ruby におけるクラスとして定義している．また，Ruby 形式によるルールの例を**図 6** に示す．図 6 は，「居間のテレビの電力が 1W より大きければ人が存在すると認識し，玄関の照明を点ける」ルールである．実装システムでは，図 2 に示したような情報をルールエンジンに登録し，図 6 に示したようなルールに基づいて電化製品を制御する．

5. まとめ

本研究では，Rete アルゴリズムを用いたルールベースの家庭内エネルギー管理システム（HEMS）の複数スマートタップによる実装について述べた．実装システムでは，電力制御のためのルールをネットワーク上のスマートタップが処理し，ルール処理およびデータ収集の負荷が分散される．また，Rete アルゴリズムに基づいてルールを処理することで，ルール処理およびデータ収集の回数が削減される．

今後の課題については，さまざまなルールや電化製品，センサデータを用いて実装システムを評価することが挙げられる．また，3.2 節で述べたスマートタップへの処理割り当て方式には実測値に基づいて担当を決定するなどの方法も考えられ，実装システムを用いた評価結果に基づき，さまざまな割り当て方式を検討することが挙げられる．

謝辞 本研究の一部は，独立行政法人情報通信研究機構（NICT）の委託研究「情報通信・エネルギー統合技術の研究開発」の助成による成果である．

参考文献

[1] 松山隆司：エネルギーの情報化とは：背景，目的，基本アイディア，実現手法，情報処理， Vol. 51, No. 8, pp. 926–933 (2010).
[2] 加藤丈和，湯浅健史，松山隆司：オンデマンド型電力制御システム，情報処理学会論文誌，pp. 1185–1198 (2013).
[3] Yoshihisa, T., Fujita, N. and Tsukamoto, M.: A Rule-Based Home Energy Management System, *Proceedings of the FTRA International Conference on Advanced IT Engineering and Management (AIM 2012)*, pp. 47–48 (2012).
[4] 義久智樹，佐野渉二，藤田直生，塚本昌彦：家庭内 EoD を用いた電力機器管理システムのための制御ルール変換方式，情報処理学会研究報告，Vol. 2012-CDS-3, No. 27, pp. 1–8 (2012).
[5] Yoshihisa, T., Fujita, N. and Tsukamoto, M.: A Rule Generation Method for Electrical Appliances Management Systems with Home EoD, *Proceedings of the IEEE Global Conference on Consumer Electronics (GCCE 2012)*, pp. 253–255 (2012).
[6] 小林重信：プロダクションシステム，情報処理， Vol. 26, No. 12, pp. 1487–1496 (1985).
[7] Forgy, C. L.: Rete: A Fast Algorithm for the Many Patterns/Many Objects Match, *Artificial Intelligence*, Vol. 19, No. 1, pp. 17–37 (1982).
[8] Shvartzshnaider, Y., Ott, M. and Levy, D.: Publish/Subscribe on Top of DHT Using RETE Algorithm, *Proceedings of the 3rd Future Internet Symposium (FIS 2010)*, pp. 20–29 (2010).

[9] Morimoto, N., Tanaka, M., Akehi, T., Yoshida, M., Yoshimizu, H., Takiyamada, M. and Kamimura, Y.: The Design and Implementation of a Smart Tap for Policy-Based Power Management, *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC) 2012 (Special Session on Ecological and Smart Home Network)*, pp. 296–300 (2012).

[10] Miranker, D. P.: TREAT: A Better Match Algorithm for AI Production Systems, *Proceedings of the 6th National Conference on Artificial Intelligence (AAAI'87)*, pp. 42–47 (1987).

[11] 木村春彦，住吉一之，小林真也，武部　幹：プロダクションシステムの直接条件照合アルゴリズム，電子情報通信学会論文誌 D-II，Vol. J77-D-2, No. 2, pp. 370–379 (1994).

[12] 南保英孝，木村春彦，広瀬貞樹：プロダクションシステムにおけるジョイン演算の順序に関する一考察，電子情報通信学会論文誌 D-II，Vol. J80-D-2, No. 10, pp. 2790–2799 (1997).

[13] Eugster, P. T., Felber, P. A., Guerraoui, R. and Kermarrec, A.-M.: The Many Faces of Publish/Subscribe, *ACM Computing Surveys*, Vol. 35, No. 2, pp. 114–131 (2003).

[14] Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R., Kaashoek, M. F., Dabek, F. and Balakrishnan, H.: Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications, *IEEE/ACM Transactions on Networking*, Vol. 11, No. 1, pp. 17–32 (2003).