

ユーザ嗜好に応じた動的なサービス構成のための リレーションシップメカニズムの設計と評価

板生 知子^{†,††} 中村 哲也[†] 松尾 真人[†]
田中 聡[†] 須田 達也^{†††} 青山 友紀^{††}

将来のコピキタスな環境では、ハードウェア、ソフトウェア、情報などあらゆるものをサービスコンポーネントとして接続するユニバーサルネットワークが浸透する。このオープンかつダイナミックなユニバーサルネットワークにおいて、ネットワーク上に分散するサービスコンポーネントをユーザニーズに応じて動的に連携させ、一連のサービスを構成するためには次の2つが課題となる。(1) ヘテロなコンポーネント間の連携を保障する。(2) ユーザ嗜好や行動パターンに応じて関連のあるコンポーネントを動的に組織化する。創発型ネットワークアーキテクチャJack-in-the-Net (Ja-Net) は、サービスコンポーネントを自律的に振る舞うCE (Cyber-Entity) としてモデル化することにより、CE間の自律的なインタラクションに基づいて有効なサービスをボトムアップ的に構成する。その際に、CEが有効な関係 (リレーションシップ) を自律的に獲得し、適応的に組織化することを可能とするリレーションシップメカニズムによって課題(1), (2)を解決する。本論文では、リレーションシップメカニズムとアプリケーションの設計と実装を通して、Ja-Netによりユーザ嗜好に応じたサービスを動的に構成する方法を示す。また、アプリケーションを用いたシミュレーション実験により、リレーションシップによって予測不能なユーザ嗜好の変化に対してもサービス構成を適応的に変更できることを示す。

Relationship Mechanism for Dynamic and User Preference-Aware Service Creation

TOMOKO ITAO,^{†,††} TETSUYA NAKAMURA,[†] MASATO MATSUI,[†]
SATOSHI TANAKA,[†] TATSUYA SUDA^{†††} and TOMONORI AOYAMA^{††}

The Jack-in-the-Net Architecture (Ja-Net) realizes a framework for autonomous and adaptive service composition in open-ended and dynamic network environments. In Ja-Net, a service is implemented by an organization of autonomous service components called *cyber-entities* that are loosely coupled based on relationships among themselves. We designed a relationship mechanism that enables cyber-entities to learn useful relationships and self-organize based on them to dynamically create services that users prefer. By implementing the relationship mechanism and cyber-entities, we showed how services that users prefer are created in Ja-Net. We also evaluated the service-adaptation capability of the relationship mechanism through a simulation-based experiment.

1. はじめに

近年、インターネット、ホームネットワーク、街頭のホットスポット、携帯端末どうして形成したアドホックネットワークなどの多様なネットワークをシームレ

スにつなぎ、その上で異なる企業、組織、個人によって提供される様々なハードウェア、ソフトウェア、情報をサービスコンポーネントとして提供する“ユニ

[†] NTT 未来ねっと研究所
NTT Network Innovation Laboratories

^{††} 東京大学大学院情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo

^{†††} カリフォルニア大学アーバイン校情報計算機科学専攻
Department of Information and Computer Science, University of California, Irvine

A part of Tatsuya Suda's research presented in this paper was supported by the National Science Foundation through grants ANI-0083074 and ANI-9903427, by DARPA through Grant MDA972-99-1-0007, by AFOSR through Grant MURI F49620-00-1-0330, and by grants from the University of California MICRO Program and Nippon Telegraph and Telephone Corporation (NTT). Tatsuya Suda also holds the title of NTT Research Professor and his NTT contact information is same as the co-authors' contact information.

“パースナルネットワーク”が現実のものとなりつつある。ユーザは、日常生活の様々なシーンにおいて、これらのサービスコンポーネントを組み合わせた一連のサービスを利用するようになるだろう。このような状況においては、サービスに対するユーザのニーズは、時間、場所など実世界の状況に連動して動的に変化する。このため、ネットワーク上に分散するサービスコンポーネントを状況に応じて適切に組み合わせることにより、ユーザニーズに応えるサービスをリアルタイムに構成することが必要となる。しかし、変化するユーザニーズを適切に把握し、これに応じたサービスを動的に構成する手法は、いまだ確立されていない。

筆者らは、オープンかつダイナミックなユニバーサルネットワーク環境において、実世界の状況やユーザ嗜好に応じて自律的に進化適応するサービスを可能とする創発型ネットワークアーキテクチャ Jack-in-the-Net (Ja-Net) を提案している^{1),2)}。Ja-Net では、ハードウェア、ソフトウェア、情報コンテンツ、ユーザなどのあらゆるシステムコンポーネントを自律的に動作する CE (Cyber-Entity) としてモデル化することにより、CE 間の自律的なインタラクションによって一連のサービスを構成する。その際に、CE は、有効な関係 (リレーションシップ) を自律的に獲得し、これに基づいて適応的に組織化することにより、サービスをボトムアップ的に構成 (創発) する。

筆者らは、これまでに、動的なインタラクションを可能とする CE と、CE の組織化を可能とするリレーションシップの基本設計を行い、プロトタイプの実装によりフィージビリティの検証を行ってきた^{3),4)}。これらの技術をシステム化するためには、次の 2 つが課題となる。(1) オープンな環境においてヘテロな CE 間のインタラクションを保障する。(2) ユーザ嗜好や行動パターンに応じて CE の自己組織化を促進する。本論文では、課題 (1)、(2) を解決するリレーションシップメカニズムを提案する。特徴は、次の 3 つである。(a) CE のサービスタイプを、CE が提供するサービスの内容とそのインタフェースによって定義し、サービスタイプを照合することによりインタラクション可能な CE を識別する。(b) リレーションシップは、CE のサービスタイプと有効度を表し、CE が他の CE のリレーションシップをテーブルとして保持することにより、CE を動的に関連づける。(c) ユーザ評価に基づいて CE が有効なリレーションシップを学習することにより、CE の自己組織化を促進する。本論文は、次の構成をとる。2 章では、リレーションシップメカニズムを実現する Ja-Net プラットフォームの設計に

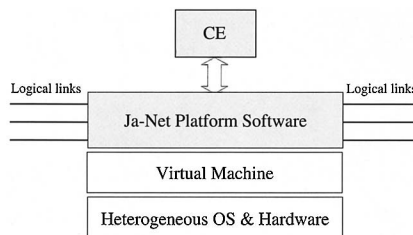


図 1 Ja-Net ノード構成

Fig. 1 Ja-Net node structure.

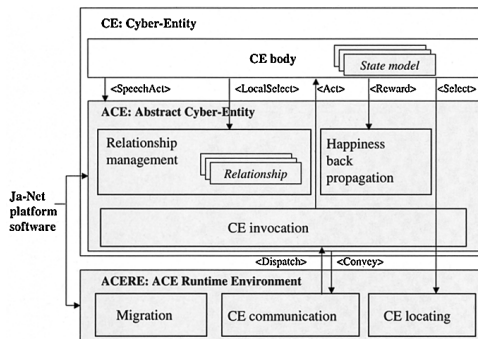


図 2 Ja-Net プラットフォームアーキテクチャ

Fig. 2 Ja-Net platform architecture.

ついて述べる。3 章では、アプリケーションの設計と実装を通して、リレーションシップメカニズムによるサービスの構成例を示す。4 章では、シミュレーション実験により、リレーションシップメカニズムによってサービスをユーザ嗜好に適応させることが可能であることを示す。5 章では、関連研究を示す。6 章で結論を述べる。

2. Ja-Net プラットフォームの設計

本章では、リレーションシップメカニズムを実現する Ja-Net プラットフォームの設計について述べる。

2.1 ノード構成

図 1 に示すように、Ja-Net は、PC や PDA などのデバイスをノードとして構成される分散システムである。各ノードは、バーチャルマシンを搭載しており、その上で Ja-Net プラットフォームソフトウェアを起動している。また、他のノードと論理リンクを形成する。CE は Ja-Net プラットフォームソフトウェア上で動作する。

2.2 プラットフォームアーキテクチャ

図 2 に、Ja-Net プラットフォームの機能構成を示す。また、図中の $\langle \rangle$ 内は機能ブロック間のメソッド名を表す。なお、図 2 に示したすべての機能ブロックは、Java J2SE1.3.1.02 上で実装済みである。

CEは、CE body, ACE(Abstract Cyber-Entity), の2つのレイヤから構成される。CE bodyは、CE単体で提供するサービスを持ち、これはCE開発者によってCEごとに実装される。ACEは、CEの共通機能群であり、主に次の3つの機能ブロックから構成される。(1)メッセージの受信を契機としてCEを起動するCE起動部(CE invocation), (2)他のCEに関するリレーションシップを管理するリレーションシップ管理部(relationship management), (3)サービスに対するユーザの評価(happiness)に基づいてリレーションシップを更新するユーザ評価伝播部(happiness back propagation)。実装上は、ACEはCEの抽象クラスであり、すべてのCEはACEを継承する。

ACERE(ACE Runtime Environment)は、CEの実行環境を提供し、主に次の3つの機能ブロックから構成される。(1)CEの移動を実現するためにACERE間でCEを移送するCE移送部(migration), (2)CE間のメッセージングを行うCE通信部(CE communication), (3)ACERE上に存在するCEを検出するCE位置管理部(CE locating)。ACEREはノードごとに存在し、近隣ノード上の他のACEREを検出し、論理リンクを形成する。また、CEの移動は、JavaのSerializableインタフェースを用いてCEを直列化し、JavaのRMIにより移動先のACEREに送付することにより実現する。

ACEとACEREを合わせてJa-Netプラットフォームと呼ぶ。CE bodyに対するJa-NetプラットフォームのAPIを表1に示す。CEの開発者は、CE bodyの中でこれらのAPIを呼ぶことによって、CEのメッセージの送受信とリレーションシップの生成(speechAct, act), インタラクショナル相手の選択(select, localSelect), ユーザ評価の伝播(reward)など、CEの組織化に必要な処理を起動する。

以降の節では、Ja-Netプラットフォームにおけるリレーションシップメカニズムの実現方法を説明する。

2.3 CE body

Ja-Netプラットフォームでは、CE bodyの処理手順を定義する状態遷移モデルクラス(図2のstate model)を提供する。また、CEの緩やかな連携を可能とするために、エージェント通信言語ACL(Agent Communication Language⁵⁾)を用いた対話型のCE間メッセージ通信をサポートする。CE開発者は、状態遷移モデルクラスを用いて、CE bodyにおけるローカルな処理と、ACLメッセージの送受信の手順を実装し、ACEに登録する。

異なるCE開発者によって提供されるCE間のイン

表1 Ja-NetプラットフォームAPI(ユーザコマンド)
Table 1 Ja-Net platform API.

メソッド名	処理内容	主な引数
speechAct	ACLメッセージの送信	ACLメッセージ, 送信側および受信側CEのサービスタイプ(オプション)
act	ACLメッセージの受信	ACLメッセージ
select	近隣ACERE上に存在するCEの検索	ACEREホップ数, サービスタイプの照合条件
localSelect	指定したCEIDを持つCEのリレーションシップ相手の検索	CEID, リレーションシップの照合条件
reward	ユーザ評価(happiness)の発行と伝播	happiness値, サービスパリスト

表2 サービスタイプのクラスフィールド
Table 2 ServiceType attributes.

フィールド名	意味
serviceName	サービス名
ipName	IP名
ontology	このサービスタイプの概念空間名

タラクションを保障するため、Ja-Netでは、CE間インタラクションにおけるメッセージのフローを“インタラクションプロトコル”(IP: Interaction Protocol)としてあらかじめ規定し、同じIPを実装したCE間でのみインタラクションを許す。また、CEのサービスタイプを、(1)CE bodyのサービス内容に関するメタデータ(サービスメタデータ), (2)CE bodyが用いるIPの仕様記述、の2つの属性により定義する。これにより、サービスタイプ中のIP仕様記述を照合することにより、インタラクション可能なCEを識別することを可能とする。

現在のJa-Netプラットフォームの実装におけるサービスタイプのクラスフィールドを表2に示す。現状では、サービスメタデータとIP仕様記述は、それぞれサービス名とIP名を表す文字列によって実装している(表2のserviceNameとipName)。サービス名およびIP名の文字列は、“.”によって区切ることにより名前空間を分割する。たとえば、オークションでチケットを売り出すサービスの場合、サービス名が“電子商取引・チケット販売”, IP名が“オークション”となる。また、サービスタイプごとに、サービス名、IP名の概念空間(オントロジ)名を指定するものとし(表2のontology), あるオントロジのもとでサービス名、IP名が一致する2つのサービスタイプは同一と見なす。

図3に、CE, 状態遷移モデル, IP, サービスタイ

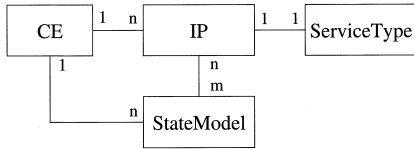


図3 インタクションに関連するクラス間の関係
Fig. 3 Relations among interaction-related classes.

表3 リレーションシップのクラスフィールド
Table 3 Relationship attributes.

フィールド名	意味
<i>ceid</i>	相手 CE の CEID
<i>serviceType</i>	相手 CE のサービスタイプ
<i>useCount</i>	このリレーションシップの選択回数
<i>strength</i>	相手 CE の有効度の指標
<i>userField</i>	ユーザの自由領域

プの関係を示す。CEは、複数のIPを使ってインタラククションすることができ、IPごとにサービスタイプを定義する。CEは、1つ以上の状態遷移モデルにより、自身のサービスおよびIPを使ったインタラククションの処理手順を定義する。このとき、1つの状態遷移モデルにより、複数のIPを使ったインタラククションを定義することができる。また、1つのIPは複数の状態遷移モデルから利用することができる。CE bodyの状態遷移モデルとサービスタイプの対応づけはシステム(ACE)が管理する。CE開発者は、状態遷移モデル中で用いるIPごとにサービスタイプを定義し、状態遷移モデルと対応づけてACEに登録する。

2.4 リレーションシップ

リレーションシップは、Ja-Netにおいて、ユーザ嗜好やサービス利用パターンに応じて関連のあるCEどうしを動的に結び付けることにより、CEの自己組織化を促進する役割を担う。

リレーションシップの実体は、相手CEに関する情報を格納するテーブルである。表3にリレーションシップのクラスフィールドを示す。Ceidは、相手の識別子(CEID)である。ServiceTypeは、相手のサービスタイプである。UseCountは、このリレーションシップの選択回数を記録する。Strengthは、リレーションシップの強度であり、自CEから見た相手CEの有効度を表す。アプリケーション固有の情報は、userFieldに記録する。CEIDとserviceTypeによって、リレーションシップを唯一に特定することができる。

CEは、複数のCEとリレーションシップを生成することができる。CEの生成したリレーションシップは、同CEのリレーションシップ管理部によって保持される。

2.5 動的なリレーションシップの生成

CEは、インタラククションを通じて、インタラククション相手のリレーションシップを生成し、テーブルとして保持する。この仕組みについて以下に説明する。

CEは、新たなCEとインタラククションを行う場合に、通信可能な範囲に存在するCEの中からインタラククション可能な相手を検索する。ACEREのCE位置管理部は、指定されたサービスタイプを持つCEを検索するselectメソッド(表1参照)を提供する。同メソッドにおいて、ホップ数を引数に指定することにより、CEを検索するACEREの範囲を指定することができる。たとえば、CE bodyから、IP名のみを指定したサービスタイプ(他のフィールドは任意)を指定してselectを呼ぶことにより、指定されたホップ数で通信可能な近隣ACERE上に存在する、特定のIPでインタラククション可能なCEを得る。

相手CEにメッセージを送信する際には、CE bodyでACLメッセージを生成し、受信側CEのCEIDとセッション情報をヘッダ部に指定する。そして、これを引数としてACEのspeechActメソッド(表1参照)を呼ぶ。このとき、speechActの引数として、ACLメッセージに加えてサービスタイプをセットした場合には、ACEによって、このメッセージ通信が完了した時点で、送信側CEおよび受信側CEのテーブルに相手CEのリレーションシップが登録される。これにより、インタラククション相手とリレーションシップを生成することができる。

他のCEからのACLメッセージを受信した場合は、ACEのCE起動部がACLメッセージを引数としてCE bodyのactメソッド(表1参照)を呼ぶことにより、受信メッセージをCE bodyに配送する。

2.6 有効なリレーションシップの学習

ユーザ嗜好に応じたサービスを構成するうえで有効なリレーションシップを学習するために、CEは、サービスを提供するたびに、サービスに対するユーザ評価に基づいて、同サービスを構成するCEのリレーションシップの有効度(strength値)を更新する。サービスに対するユーザ評価値はhappiness変数によって表される。happinessは、1, 0, -1のいずれかの値をとり、サービスに対する満足度に応じて次の値が設定される。“満足”(happiness値1)、“ニュートラル”(happiness値0)、“不満足”(happiness値-1)。なお、happiness値は、人間のユーザが決定しユーザCEに通知する方法や、ユーザCEがユーザの行動を観測することにより自動的に判定する方法などがあるが、現在の実装ではユーザによる明示的な通知を行う前者の

方法を採用している。

以下に、*happiness*の伝播と、リレーションシップ強度の更新方法を説明する。

2.6.1 ユーザ評価の伝播

*Happiness*を伝播するパスを生成するために、各CEのユーザ評価部は、サービス提供時にサービスに関係したすべてのCEのインタラクションのパスをサービスパスリストに記録する。具体的には、ユーザ評価伝播部は、インタラクション時にCEが*speechAct*メソッドによってメッセージを送信するタイミングでエントリを生成し、サービスパスリストに追加する。このエントリは、送信側CEのCEIDとサービスタイプ、受信側CEのCEIDとサービスタイプ、の組から構成され、メッセージの送受信を行った順番でサービスパスリストに格納される。サービスパスリストは、ACLメッセージの中に格納され、次々とCE間を渡り、最終的にサービスを受けたユーザCEに渡される。ユーザCEが、サービスパスリストと*happiness*値を引数として*reward*メソッド(表1参照)を呼びと、各CEのユーザ評価伝播部が連携してサービスパスリスト上のすべてのCEに1度ずつ*happiness*を伝播させる。

2.6.2 リレーションシップ強度の更新

CEは、*happiness*値に基づいてリレーションシップ強度を次のように変更する。*Happiness*値が1(-1)の場合は、インタラクション相手に対するリレーションシップの*strength*値を増加(減少)させる。*Happiness*=0の場合には変更しない。これにより、ユーザ評価が高い(低い)リレーションシップの場合は*strength*値が大きく(小さく)なる。

リレーションシップ強度算出関数は、システム生成時に組み込まれる。その際に、上記性質を満たす関数であれば任意の関数を設定することができる。現在の実装では、以下に示す式(1)を用いている。

$$S = \frac{P}{A} \quad (1)$$

ただし、*S*は*strength*値、*P*は*happiness*値1を受けた回数、*A*は*happiness*値1、-1のいずれかを受けた回数の累計である。また、*P*、*A*の初期値は1(すなわち、*S*の初期値は1)とする。この場合、*S*の値は負の0から1の間の値をとる。*Happiness*の受信回数が十分大きくなると、*strength*値は*happiness*値1を受信する確率に近づく。

2.7 CEの自己組織化

Ja-Netでは、CEがリレーションシップに基づいてインタラクション相手を選択することにより、CEの自己組織化を促進する。

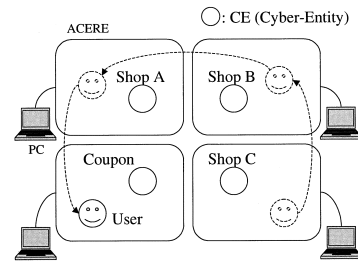


図4 アプリケーションの動作環境
Fig. 4 Application environment.

リレーションシップ管理部は、CEがリレーションシップ相手の中からインタラクション相手を選択するための*localSelect*メソッド(表1参照)を提供する。*LocalSelect*は、引数で指定したCEの持つリレーションシップの中から、指定したサービスタイプに一致するサービスタイプを持つリレーションシップを検索し、そのリレーションシップ相手を返す。たとえば、リレーションシップのサービスタイプと*strength*の閾値を照合条件として*localSelect*を呼び出すことにより、過去にインタラクションしたことのあるCEの中から、特定のサービスタイプを持ち、かつ*strength*値が閾値以上のCEを得る。

このとき、CE bodyにおいて、“*localSelect*でCEを検索した結果、有効な(*strength*値が閾値以上の)リレーションシップ相手が多く得られた場合にはこれらのCEの中からインタラクション相手を選択し、そうでない場合には*select*によって新たなCEを検索する”，というインタラクション相手の選択論理を実装することにより、適応的にサービスを構成することができる。

3. リレーションシップによるサービス構成例

Ja-Netのアプリケーション例として、ユーザの嗜好に応じて動的に商品の割引クーポンを発行するサービス(クーポンサービス)を構成する複数のCEを設計し、Ja-Netプラットフォーム上で実装した。この結果、CEが自律的にインタラクションを行うことによって動的にクーポンサービスを構成することができた。また、ユーザ評価に基づいてCE間のリレーションシップ強度を更新することにより、生成されるクーポンがユーザ評価を反映して変化することを確認した。

上記アプリケーションの構成と動作環境を図4を用いて説明する。本アプリケーションは、クーポンを生成するクーポンCE(“Coupon”)、異なる商品を販売する3つの店舗CE A, B, C(“Shop”), ユーザCE(“User”)から構成される。クーポンCEおよび

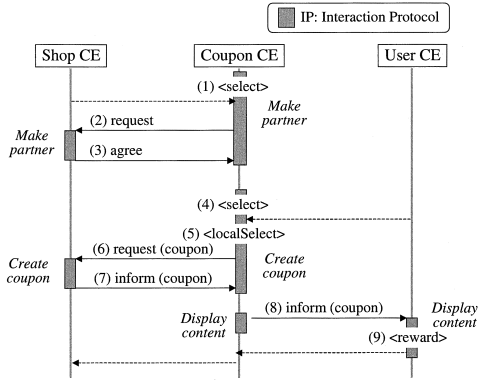


図5 アプリケーションの動作シーケンス Fig. 5 Application sequence.

3つの店舗 CE は隣接する異なる ACERE 上で動作する。また、これらの ACERE は、異なる PC 上で動作する。ユーザ CE は ACERE 間をランダムに移動する(本アプリケーションでは、ユーザ CE はユーザの PDA 上に乗っていることを想定し、ユーザ CE の移動により、人間のユーザの行動をシミュレートする)。

本アプリケーションの動作シーケンスを図5を用いて説明する。ただし、すべての店舗 CE は同じ動作をするため、図5では1つの店舗 CE だけを示す。また、図5のメッセージラベルのうち、<>で囲まれたものは Ja-Net プラットフォームの API(表1 参照)のメソッド名を表し、それ以外は ACL メッセージを表す。なお、アプリケーションの初期状態では、すべての CE の間でリレーションシップはいっさい生成されていない。

(1) アプリケーションが起動した直後に、クーポン CE は店舗 CE を検索し(“(1)<select>”),この結果、3つの店舗 CE A, B, Cが見つかる。クーポン CE は、各店舗 CE と“提携依頼”IP(“make partner”)を使ってインタラクションすることにより、相手の店舗と提携関係を結ぶ(“(2) request”および“(3) agree”)。このインタラクションにより、クーポン CE と各店舗 CE の間にリレーションシップが生成される。

(2) クーポン CE は、自 ACERE 上のユーザ CE を検索し(“(4)<select>”),新しいユーザ CE を検出したタイミングでリレーションシップを検索し(“(5)<localSelect>”),リレーションシップ相手(店舗 CE A, B, C)の中から1つを選択する。そして、選択した店舗 CE と“クーポン生成”IP(“create coupon”)を使ってインタラクションすることにより、同店舗のクーポンを生成する(“(6) request (coupon)”および“(7) inform (coupon)”)。続いて、先に検出したユー

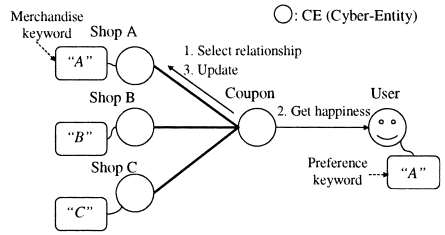


図6 シミュレータの動作概要 Fig. 6 Overview of simulator behavior.

ザ CE と“画面表示”IP(“display content”)を使ってインタラクションすることにより、クーポンをユーザ CE に送付する(“(8) inform (coupon)”)。

(3) ユーザ CE は、受信したクーポンをユーザの PDA 上に表示し、これを見た人間のユーザが商品を購入するか否かを決定し、ユーザ CE に通知する。ユーザ CE は、商品を購入する(しない)場合に happiness の値を 1 (-1) を発行する(“(9) <reward>”)。Happiness の通知はクーポン CE, 店舗 CE の間を伝播し、happiness 値が 1 (-1) の場合には、クーポン CE と店舗 CE のリレーションシップが強化(弱化)される。

4. ユーザ適応能力の検証

3章で示したクーポンサービスにおいて、ユーザの大半が、店舗 A のクーポンよりも店舗 B のクーポンを利用する場合には、クーポン CE は店舗 CE B と優先的に連携することが望ましい。しかし、ユーザの嗜好が動的に変化する場合には、これに応じて連携する店舗を切り替えることが必要である。以上の考えに基づき、3章で示したクーポンサービスを用いて、以下の2つの観点からリレーションシップによる適応能力を評価した。

E1: ユーザ嗜好に応じたサービスを動的に構成することができる。

E2: ユーザ嗜好が変化した場合に、これに追従してサービスの構成を変化させることができる。

なお、本実験は、3章で示したシステムとは別に、クーポンサービスのリレーションシップを模擬するシミュレータプログラムを作成して行った。

4.1 シミュレータの概要

図6を用いて実験で用いるシミュレータプログラムの動作概要を説明する。店舗 CE は、それぞれの店舗の商品を表す商品キーワード(A, B, C)(“merchandise keyword”)を1つ持つ。一方、各ユーザ CE は、商品に対する嗜好を表す嗜好キーワード(“preference keyword”)を1つ持つ。ユーザ CE の

嗜好キーワードは、店舗 CE の商品キーワードのどれか 1 つと等しい。

本シミュレータでは、シミュレーション開始時に、クーポン CE は各店舗に対するリレーションシップを 1 つずつ生成する。また、単位時間あたりに 1 つのユーザ CE を生成し、次の処理を行う。クーポン CE は、リレーションシップに基づいて 3 つの店舗から 1 つを選択し (“1. select relationship”), その商品キーワードをユーザに渡して *happiness* 値を取得する (“2. get happiness”). ユーザ CE は、商品キーワードが自身の嗜好キーワードと一致する (一致しない) 場合に *happiness* 値 1 (-1) を返す。クーポン CE は、受信した *happiness* 値に基づいてインタラクショナル相手の店舗 CE に対するリレーションシップの強度を更新する (“3. update”).

4.2 シミュレーションの設定

本シミュレーションでは、ユーザ嗜好のトレンド、リレーションシップ強度算出関数、リレーションシップ選択論理を、それぞれ以下に述べるように設定した。

4.2.1 ユーザ嗜好のトレンド

本シミュレーションでは、ユーザの嗜好に何らかのトレンドがあることを想定し、ユーザ CE の持つ嗜好キーワードの出現確率に偏りを与える。この偏りのレベルは、すべてのユーザの嗜好キーワードの集合のエントロピー値によって定量的に示すことができる。エントロピー H の算出式は、以下によって与えられる。

$$H = \sum_A -P_a \log_2 P_a \quad (2)$$

ただし、 P_a は、嗜好キーワード空間 A におけるキーワード a の出現確率である。式 (2) において、 H の値は右辺の多項式の各項 ($-P_a \log_2 P_a$) の値がすべて等しい場合に最大となることから、嗜好キーワードの出現確率が一樣な (偏りが無い) 場合に、エントロピー値が最大となる。すなわち、エントロピー値が高い (低い) ほど、ユーザの嗜好キーワードの偏りが小さく (大きく) なる。

表 4 に、本実験で用いる 3 種類の嗜好キーワード (A, B, C) の出現頻度の比とエントロピー値の関係を示す。たとえば、エントロピー値が 1.57 の場合は、クーポンを受信したユーザ 10 人のうち、4 人が嗜好キーワード A , 3 人がそれぞれ嗜好キーワード B, C を持つことを表す (この場合、嗜好の偏りは小さいといえる)。一方、エントロピー値が最も低い 0 の場合は、クーポンを受信したすべてのユーザ (10 人中 10 人) が嗜好キーワード (A) を持つことを表す (この場合、嗜好の偏りは大きいといえる)。

表 4 ユーザ嗜好キーワードのエントロピー
Table 4 The entropy of user preference keywords.

エントロピー値 (H)	嗜好キーワード出現頻度の比					
	変更前 (before)			変化後 (after)		
	A	B	C	A	B	C
1.57	4	3	3	3	3	4
1.49	5	3	2	3	2	5
1.37	6	2	2	2	2	6
1.16	7	2	1	2	1	7
0.92	8	1	1	1	1	8
0.72	8	2	0	2	0	8
0.47	9	1	0	1	0	9
0	10	0	0	0	0	10

また、実験においてユーザの嗜好を変化させる場合は、エントロピー値を変えずに、各嗜好キーワードの出現頻度の比だけを変更する。たとえば、エントロピー値 1.49 のままで、ユーザの嗜好を変化させる場合は、 A, B, C の出現頻度の比を 5 : 3 : 2 (“変更前”) から 3 : 2 : 5 (“変化後”) に変化させる。

4.2.2 リレーションシップ強度算出

リレーションシップ強度算出関数は、2.6.2 項に示した式 (1) を用いる。

4.2.3 インタラクショナル相手の選択論理

クーポン CE は、リレーションシップの強度を基準として、クーポンを生成する店舗 CE を選択する。リレーションシップ強度は、過去に受けたユーザ評価の履歴を反映して強化 (弱体化) されているため、ユーザ嗜好があまり変化しない場合においては、リレーションシップ強度の最も強い店舗を選択することが有効と考えられる。しかし、ユーザ嗜好が途中で変化する場合には、過去の履歴が有効でなくなるため、上記の方法ではユーザ嗜好の変化への追従が遅くなると考えられる (これは、過去に評価の高かった店舗のリレーションシップ強度が、ユーザから *happiness* 値 -1 を受け続けることにより弱体化されることによって、新たに評価の高くなった店舗のリレーションシップ強度を下回るまで、新たに評価が高くなった店舗を選択することができないためである)。これに対応するためには、現時点でリレーションシップが強くない店舗でもある割合で選択することによって、ユーザ嗜好の変化があった場合に、新たに評価の高くなった店舗に対するリレーションシップを速やかに強化することが有効であると考えられる。

以上の考えに基づき、本実験では、クーポン CE によるインタラクショナル相手の選択論理として、以下の 3 つを比較する。

M1 リレーションシップ強度が一番高いものを選択する (これにより、過去の履歴において最も有効で

あった CE を選択する)。

M2 リレーションシップ強度に比例した確率により選択する(たとえば, 2つのリレーションシップ強度の比が 7:3 の場合は, 0.7, 0.3 の確率でどちらか 1つを選択する)。

M3 リレーションシップ強度に関係なく, ランダムに 1つを選択する(本手法は, リレーションシップを用いない手法として比較のために用いる)。

4.3 評価指標

リレーションシップの有効性は, クーポン CE が生成したクーポンのヒット率 B を指標として評価する。このとき,

$$B = \frac{C}{T} \quad (3)$$

であり, C は, クーポン CE が相手の店舗 CE のクーポン生成した場合に, ユーザ CE から *happiness* 値 1 を受信した回数, T は上記ユーザ CE から *happiness* 値 1 または -1 を受信した回数の総数である。ヒット率は, 最も最近受信した 20 個の *happiness* を対象として計算する。リレーションシップにより, ユーザの嗜好に応じたクーポン生成が行われている場合は, ヒット率 B の値が上昇することが期待される。

4.4 シミュレーション結果

表 4 に示したエントロピー値 0.92, 1.16, 1.37, 1.49, 1.57 の場合において, 選択方法 M1, M2, M3 を用いてクーポンを生成した場合の, クーポンのヒット率(式(3))を, 以下の 2つの方法で測定した。

測定 1: クーポン CE のリレーションシップは初期状態 ($S = 1$) から開始し, クーポンを 200 回生成する(200 人のユーザに対してクーポンを表示して, *happiness* を得る)。ユーザの嗜好キーワードは, 表 4 の“変更前”(before)を用いる。

測定 2: 測定 1 の終了後に, 測定 1 のリレーションシップの最終状態から, ユーザの嗜好キーワードのみを表 4 における“変更後”(after)に変更し, クーポンを 200 回生成する。

図 7~11 に, エントロピー値がそれぞれ 0.92, 1.16, 1.37, 1.49, 1.57 の場合におけるヒット率の推移を, 選択論理 M1, M2, M3 ごとに示す。また, 図 12, 13 に, エントロピー値が 0.92, 選択論理 M1, M2 の場合の, 図 14, 15 に, エントロピー値が 1.37, 選択論理 M1, M2 の場合のリレーションシップ強度の推移を, 店舗 A (“Shop A”), B (“Shop B”), C (“Shop C”) ごとに示す。上記各図において, 左側のグラフは測定 1, 右側のグラフは測定 2 におけるヒット率を示す。また, グラフの横軸は, クーポン生成回数(すな

わち, ユーザから *happiness* を受信した回数)である。なお, 各データは, 10 回の試行の平均である。

4.5 考察

以下に, シミュレーション結果を分析し, 考察を述べる。

(1) 測定 1 のヒット率とエントロピーの関係

測定 1 では, エントロピー値が 1.49 以下の場合には, M1, M2 が M3 よりも高いヒット率を達成していることが確認できる。この傾向は, エントロピー値が低い(ユーザの嗜好の偏りが大きい)ほど顕著である。一方, エントロピー値が 1.09 の場合は, ユーザの嗜好はほぼ一様に分布しているため, リレーションシップの強度を考慮してもランダムに選択した M3 の場合と同等の効果しかない。また, 図 12, 13 および図 14, 15 の左側(測定 1)のグラフから, M1, M2 は, エントロピー値 0.92, 1.37 の両方の場合において, 店舗 A に対するユーザ嗜好の偏りを反映して, 店舗 A のリレーションシップ (“Shop A”) が最も強くなっていることが確認できる。このことから, ユーザの嗜好がある程度偏っている状況では, リレーションシップの強度がユーザの嗜好を反映して変化し, この結果ヒット率を上昇させることが分かる。

(2) 測定 2 のヒット率とエントロピーの関係

測定 2 では, エントロピー値が 1.49 以下の場合には, 嗜好の変化に対して M1, M2 が追従していることが確認できる。この傾向は, エントロピー値が低い(ユーザの嗜好の偏りが大きい)ほど顕著である。一方, エントロピー値が 1.57 の場合は, 測定 1 の理由と同様に, ユーザ嗜好変化への追従に関してはランダムに選択した M3 の場合と同等の効果しかない。また, 図 12, 13 および図 14, 15 の右側(測定 2)のグラフから, M1, M2 は, エントロピー値 0.92, 1.37 の両方の場合においてユーザ嗜好の変化に応じて, 店舗 A のリレーションシップ強度 (“Shop A”) が弱体化され, 店舗 C のリレーションシップ強度 (“Shop C”) が強化されていることが確認できる(ただし, エントロピー値 1.37 の場合は変更のスピードが遅い)。このことから, リレーションシップ強度がユーザ嗜好の変化に追従して変化することにより, 新しいユーザ嗜好に応じてヒット率を上昇させることができることが分かる。

(3) 測定 1 における M1 と M2 の比較

測定 1 では, M1 の方が M2 よりもヒット率が高い。これは, 測定 1 においてはユーザ嗜好の偏りが一定であるため, 過去において最も有効であったリレーションシップを毎回選択する M1 の方が, 確率的に選択する M2 よりも有効に働くことを示す。

(4) 測定 2 における M1 と M2 の比較

測定 2 では、M2 の方が M1 よりも少ない回数でヒット率が上昇している。すなわち、ユーザ嗜好の変化に対する追従のスピードが速い。これは、リレーションシップ強度が現在のユーザ嗜好を反映するように正しく修正されるためにかかる期間と関係する(すなわち、過去のユーザ嗜好において人気のあった店舗 A のリレーションシップ強度 (“Shop A”) を弱化し、新しいユーザ嗜好において人気のある店舗 C のリレーションシップ強度 (“Shop C”) を強化すること)。

- M1 では、店舗 A のリレーションシップ強度 (“Shop A”) が店舗 C のリレーションシップ強度 (“Shop C”) より小さくなるまで、店舗 C を選択することができず、この間リレーションシップ強度を修正することができない。たとえば、図 12 の右側のグラフ (測定 2) では、ユーザ嗜好の変化後、クーポンを 150 回程度発行するまで、店舗 CE を選択することができない。図 7 の右側のグラフ (測定 2) から、この間 M1 のヒット率が改善されていないことが分かる。
- これに対して、M2 ではつねにすべてのリレーションシップを確率的に選択しているため、ユーザ嗜好の変化直後から店舗 C を選択することができる。このため、変化に対して速やかにリレーションシップ強度を修正することができる。たとえば、図 13 の右側のグラフ (測定 2) では、ユーザ嗜好が変化した直後から店舗 A のリレーションシップ強度 (“Shop A”) の弱化、店舗 C のリレーションシップ強度 (“Shop C”) の強化が始まる。これに対応して、図 7 の右側のグラフ (測定 2) では、クーポンを 40 回発行した辺りから M2 のヒット率が改善し始めている。

今回の実験により、ユーザ嗜好の偏りがある場合には、リレーションシップを用いることによって適応的にサービスを構成し、またユーザ嗜好の変化に応じてサービスを再構成することが可能であることが分かった (E1, E2 を充足する)。

上記 (1)~(4) の分析に示すように、リレーションシップ選択論理 M1, M2 は、サービスの適応的な構成を行ううえで、それぞれ異なる特徴を持っている。M1 は、ユーザの嗜好の偏りがある程度大きく、その変化が比較的少ない場合に有効である。一方、M2 は、確率的な要素を含むため、ユーザの嗜好の偏りがある程度大きく、さらにそれが動的に変化する場合に有効である。このため、E1 に対する要求が強く、E2 に対する要求が無視できる環境では、M1 を用いてサービス

を構成することが有効である。このようなサービスの例としては、ホームネットワークにおける家電の連携サービス、社内のワークフロー、観光地でのガイドなど、特定の目的にカスタマイズしたサービスがある。一方、E1, E2 をともに考慮する必要がある環境では、M2 を用いることが望ましい。このようなサービスの例としては、不特定多数のユーザが訪れる繁華街やレストランのホットスポットなどにおける、比較的短期間で変化する流行を反映したサービスがある。

ただし、今回の実験データは、ユーザ嗜好を比較的極端に変化させ (表 4 の “変化前”, “変化後”), またユーザ嗜好が変化する周期を 200 回 (クーポン発行回数) とする、ある限定されたケースにおけるものである。よって、ユーザ嗜好を緩やかに変化させた場合や、変化の周期を短くした場合のリレーションシップの振舞いについてはさらなる実験が必要である。

このため、CE 開発者は、サービスの種類やサービスを提供する環境に応じて、適切なリレーションシップ関数、およびインタラクション相手選択論理を設計することが重要である。

現在、世の中で行われている通常のクーポンサービスでは、クーポンサービスの提供者が提携先の店舗をあらかじめ選択し、その店舗のクーポンを固定的に生成する。この場合、どの店舗と提携するかの決定は人間が行うため正確な予測は困難である。また、提携相手をユーザの嗜好に応じて動的に切り替えることはできない。これを本実験に置き換えると、通常のクーポンサービスは、クーポン CE が 1 つの店舗 CE と固定的に連携するケース、あるいはクーポン CE が 3 つの店舗 CE の中からランダムに 1 つを選択して連携する M3 のケースに相当する。後者のケースで考察を進めると、実験結果から、Ja-Net 方式である M1, M2 の方が、M3 より性能が高い。一方、前者のケースでも、クーポン CE が偶然最も人気の高い店舗を選択しない限り、Ja-Net と同等の性能を達成することはできない。また、その場合でも、ユーザの嗜好が変化した場合には追従できない。以上から、Ja-Net の方式のクーポンサービスが通常のクーポンサービスよりユーザの嗜好を反映するために有効であるということができる。

5. 関連研究

Ja-Net は、リレーションシップによってユーザの嗜好を学習し、これに適応して動的にサービスを構成する機能を内包している点で、従来の Web ベースのサービスアーキテクチャと異なる。Web のアーキテク

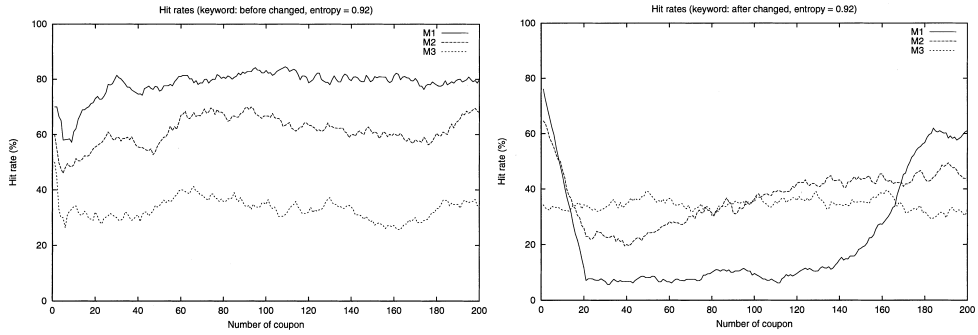


図7 エントロピー値 0.92 におけるヒット率の推移 (左: 測定 1, 右: 測定 2)
Fig. 7 Hit rate at entropy 0.92.

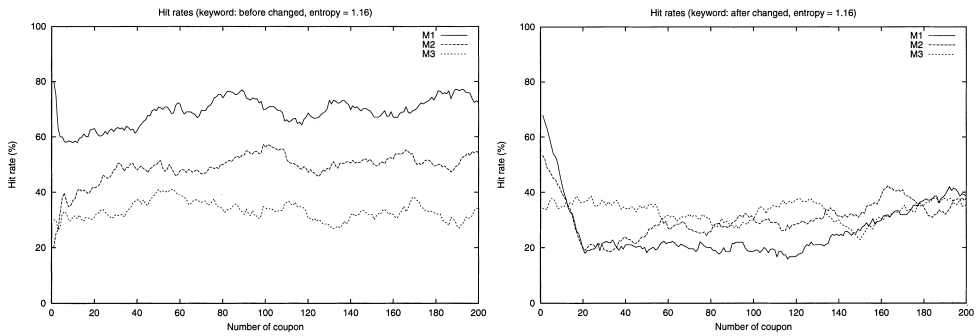


図8 エントロピー値 1.16 におけるヒット率の推移 (左: 測定 1, 右: 測定 2)
Fig. 8 Hit rate at entropy 1.16.

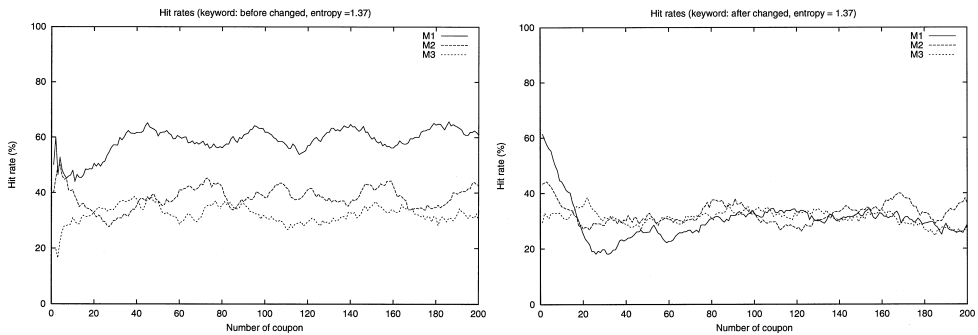


図9 エントロピー値 1.37 におけるヒット率の推移 (左: 測定 1, 右: 測定 2)
Fig. 9 Hit rate at entropy 1.37.

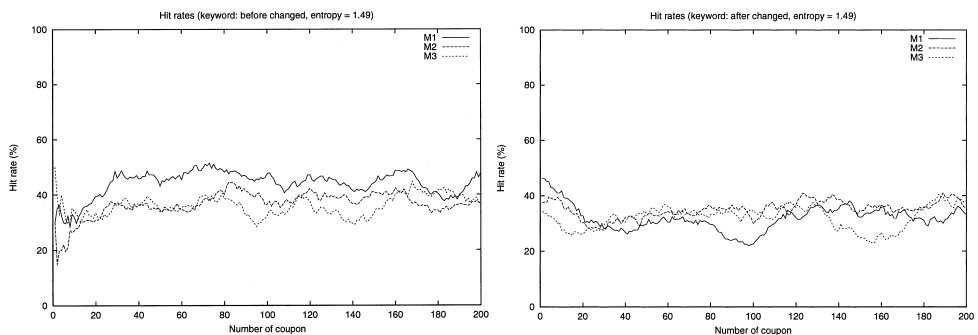


図10 エントロピー値 1.49 におけるヒット率の推移 (左: 測定 1, 右: 測定 2)
Fig. 10 Hit rate at entropy 1.49.

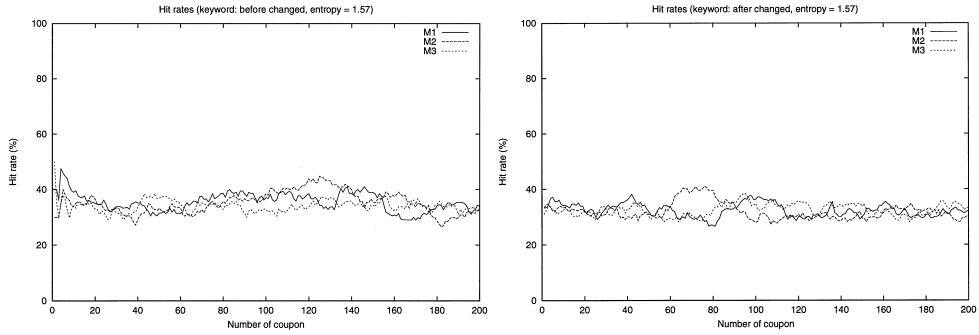


図 11 エントロピー値 1.57 におけるヒット率の推移 (左: 測定 1, 右: 測定 2)
 Fig. 11 Hit rate at entropy 1.57.

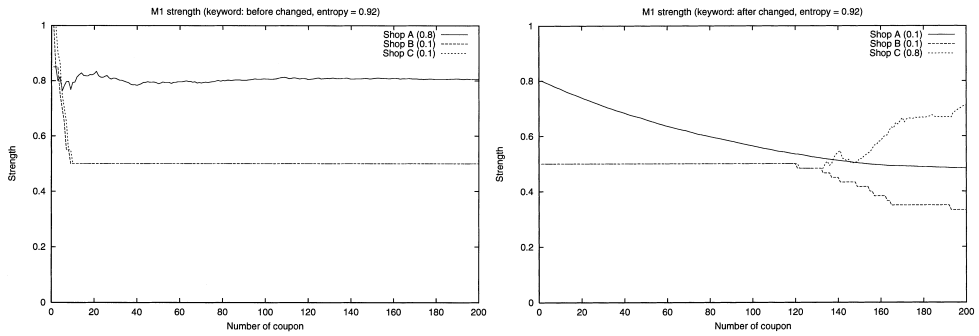


図 12 選択論理 M1, エントロピー値 0.92 におけるリレーションシップ強度の推移 (左: 測定 1, 右: 測定 2)
 Fig. 12 Relationship strengths based on M1 at entropy 0.92.

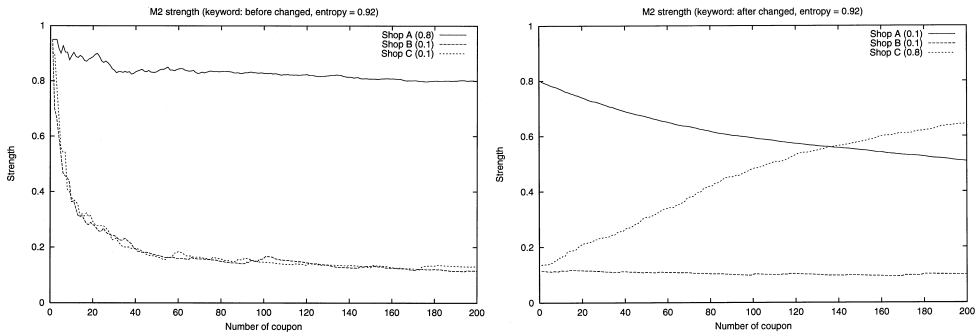


図 13 選択論理 M2, エントロピー値 0.92 におけるリレーションシップ強度の推移 (左: 測定 1, 右: 測定 2)
 Fig. 13 Relationship strengths based on M2 at entropy 0.92.

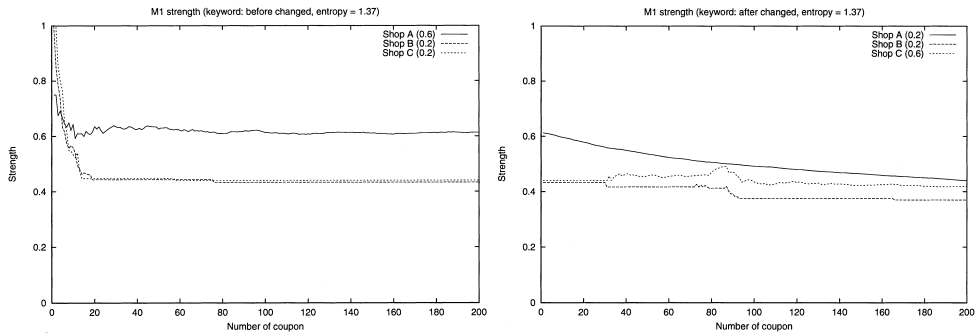


図 14 選択論理 M1, エントロピー値 1.37 におけるリレーションシップ強度の推移 (左: 測定 1, 右: 測定 2)
 Fig. 14 Relationship strengths based on M1 at entropy 1.37.

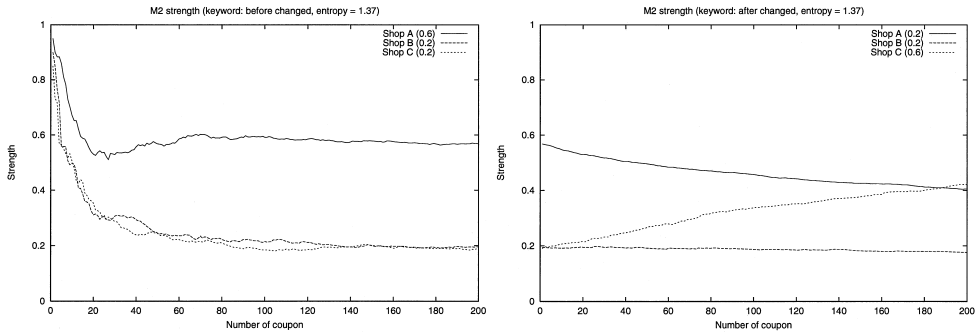


図 15 選択論理 M2, エントロピー 1.37 におけるリレーションシップ強度の推移 (左: 測定 1, 右: 測定 2)

Fig. 15 Relationship strengths based on M2 at entropy 1.37.

チャの発展形として業界が推進する Web Services⁷⁾ は、異なる企業の提供するサービスをユーザ要求に基づいて動的に連携させて一連のサービスを構成するためのフレームワークである。Ja-Net では、Web サービスの機能に加えて、リレーションシップメカニズムにより、明示的なユーザ要求がなくても適切なサービスを自動的に構成することが可能となる。

Semantic Web^{8),9)} は、Web 上のコンテンツに意味情報や関係情報などのメタデータを付加することにより、高度な情報処理 (たとえば、情報検索) を実現できるようにする技術である。Semantic Web では、Web コンテンツの背後にある論理構造や概念空間 (オントロジ) の記述と、これを利用した情報処理の論理がポイントとなるが、現段階ではまだ概念が先行しており仕様化が待たれる状態である。これに対して、Ja-Net は、Web コンテンツだけでなく一般のサービスも対象とし、サービスどうしを関連づける論理 (Semantic Web における高度な情報処理の具体的例) をフレームワーク化している点で、優位性がある。

エージェント間のインタラクションによってサービスを構成するフレームワークとしては、Hive¹⁰⁾ や Bee-gent¹¹⁾ がある。Hive では、エージェント間のインタラクションはメソッド呼び出しによって行う。このため、各エージェントは自身がインタラクションする可能性のある CE のインタフェースをあらかじめ実装する必要があり、インタラクションの柔軟性が制限される。これに対して、Ja-Net では、CE 間は ACL メッセージによる対話的な通信を行うため、動的にインタラクションを行うことが可能である。一方、Bee-gent では、仲介エージェントが複数のエージェント間のメッセージフロー (すなわち、IP: Interaction Protocol) を集中管理し、エージェント間のインタラクションをコーディネートすることにより、IP の開発・管理を容易化する。これに対して、Ja-Net では、IP を CE2 者

間でのみ定義することにより IP の複雑化を抑制する。さらに、CE が有効なリレーションシップを自律的に獲得することにより、ボトムアップ的にサービスを構成することから、Bee-gent の仲介エージェントによるトップダウン的な方法では構成できない新しいサービスを創発できる可能性がある。

Web 上のサーバによってログを収集、分析することにより、ユーザの嗜好や行動パターンを検出する技術として、データマイニングアルゴリズム¹²⁾ や、AwarenessNet^{13),14)} がある。これらのアルゴリズムを適用したシステムでは、たとえば Web ユーザのブックマークを分析することによって、ユーザの行動パターンに応じた Web ページ間の関係を抽出し、これに基づいて適応サービスを構成することが考えられる。しかし、この方法では、大量のログを収集するために生じるタイムラグにより、動的に変化する環境においては、分析の結果と現状が一致しなくなる危険性がある。これに対して、Ja-Net では、サービスを構成するたびにユーザ評価に基づいてリレーションシップをインクリメンタルに更新するため、現状のユーザ嗜好をタイムリに把握したサービスを構成することが可能である。

Sun の JXTA¹⁵⁾ は、P2P ネットワーク上で動作する分散アプリケーションの開発を支援するためのプラットフォームであり、P2P ネットワークへのノードの参加・離脱、ノードの発見、サービスのコミュニティ (グループ) 管理などに必要なプロトコル群を規定する。JXTA のノードは、Ja-Net の ACERE とその上で動作する CE に対応づけられる。このため、JXTA は、Ja-Net において ACERE 間の P2P ネットワークを構成・管理するツールとして有効である。

Ja-Net プロジェクトは、カリフォルニア大学アーバイン校 (UCI) の Bio-Net プロジェクト^{16),17)} と連携しながら研究を進めている。Bio-Net では、主に CE の振舞いの創発を、Ja-Net では CE のリレーション

シップによるサービスの創発をそれぞれターゲットとしている。

6. 結 論

オープンかつダイナミックなユニバーサルネットワーク環境において、自律的なサービスコンポーネントである CE (Cyber-Entity) が、ユーザ嗜好に応じて有効なリレーションシップを自律的に獲得することにより、適応的に組織化することを可能とする、Ja-Net のリレーションシップメカニズムを提案した。システムおよび CE 開発者に提供する API を設計し、実装によりフィージビリティを確認した。また、アプリケーションを用いたシミュレーション実験により、サービスをユーザ嗜好に適応させるためには、予想されるユーザ嗜好の変動パターンに応じてリレーションシップ強度算出回数やインタラクション相手の選択論理を適切に設計することが重要であることが分かった。

今後の課題としては、他技術に対するリレーションシップメカニズムの有効性の評価がある。たとえば既存のデータマイニング技術との比較・評価などを行う必要がある。また、オープンな環境で提供される CE 間のインタラクションを円滑に行うためには、サービスタイプで指定されているサービス名や IP 名が厳密に一致しなくても、仕様上の整合性が確認できればインタラクションを許すなど、CE 間の緩やかな連携を許す仕組みについて検討する必要がある。また、異なる CE 開発者が定義した IP を他者に公開したり、IP のバージョン間の対応を管理するなど、IP の運用方法を確立することが必要である。さらに、Ja-Net が変化し続ける環境やユーザの嗜好に真に適応するためには、サービス解消方法や、ユーザの利便性を考慮したサービス提供方法など、サービスのライフサイクルをモデル化するための検討が必要である。

参 考 文 献

- 1) Suda, T., Itao, T., Nakamura, T. and Matsuo, M.: A Network for Service Evolution and Emergence: Jack-in-the-Net, *Journal of the IEICE*, B, Vol.J84-B, No.3, pp.310-320 (2001).
- 2) 須田達也, 松尾真人, 板生知子, 中村哲也, 今田美幸, 大塚卓哉, 田中 聡: アプリケーション創発のための適応型ネットワークアーキテクチャ: Ja-Net, *情報処理*, Vol.43, No.6 (2002).
- 3) Itao, T., Nakamura, T. and Matsuo, M., Suda, T. and Aoyama, T.: Service Emergence based on Relationship among Self-Organizing Entities, *Proc. IPSJ/IEEE SAINT2002*, Nara, Japan (Jan. 2002). (Best Paper)
- 4) Itao, T., Nakamura, T., Matsuo, M., Suda, T. and Aoyama, T.: Adaptive Creation of Network Applications in the Jack-in-the-Net Architecture, *Proc. IFIP Networking 2002*, Pisa, Italy (May 2002).
- 5) FIPA-ACL web site. <http://www.fipa.org>
- 6) 長尾智晴: 最適化アルゴリズム, 昭晃堂.
- 7) Web services web site. <http://www.webservices.org/>
- 8) Semantic web web site. <http://www.w3.org/2001/sw/>
- 9) Bemers-Lee, T., Hendler, J. and Lassila, O.: The Semantic Web, *Scientific American* (May 2001).
- 10) Minar, N., Gray, M., Roup, O., Krikorian, R. and Maes, P.: Hive: Distributed Agents for Networking Things, *Proc. ASA/MA '99* (Aug. 1999).
- 11) Kawamura, T., Tahara, Y., Hasegawa, T., Ohsuga, A. and Honiden, S.: Bee-gent: Bonding and Encapsulation Enhancement Agent Framework for Development of Distributed Systems, *Journal of the IEICE*, D-I, Vol.J82-D-I, No.9 (1999).
- 12) Agrawal, R., Imielinski, T. and Swami, A.: Mining Association Rules between Sets of Items in Large Databases, *Proc. SIGMOD 1993*, pp.207-216 (1993).
- 13) Kawamura, T., Arinaga, K. and Sato, M.: A Study on Communication Distance to Realize Awareness Services, Technical Report of IEICE, IN96-32 (1996).
- 14) AwarenessNet web site. <http://www.ntt-east.co.jp/awarenessnet>
- 15) JXTA web site. <http://www.jxta.org>
- 16) The BNA project web site. <http://netresearch.ics.uci.edu/bionet>
- 17) Wang, M. and Suda, T.: The Bio-Networking Architecture: A Biologically Inspired Approach to the Design of Scalable, Adaptive, and Survivable/Available Network Applications, *Proc. IPSJ/IEEE SAINT*, San Diego, USA (Jan. 2001).

(平成 14 年 7 月 15 日受付)

(平成 14 年 11 月 5 日採録)



板生 知子(正会員)

平成6年東京工業大学工学部電気電子工学科卒業。平成8年米国スタンフォード大学大学院コンピュータサイエンス修士課程修了。同年日本電信電話(株)入社。ネットワークサービスシステムの研究開発に従事。同社未来ねっと研究所研究員。電子情報通信学会会員。



中村 哲也(正会員)

平成8年京都大学理学部物理学科卒業。平成10年東京大学大学院工学系研究科システム量子工学専攻修士課程修了。同年日本電信電話(株)入社。ネットワークサービスシステムの研究開発に従事。同社未来ねっと研究所研究員。



松尾 真人(正会員)

昭和61年京都大学工学部精密工学科卒業。昭和63年同大学大学院修士課程修了。同年日本電信電話(株)入社。インテリジェントネットワーク、分散処理ネットワークシステム、ネットワークサービスシステムの研究開発に従事。同社未来ねっと研究所主任研究員。電子情報通信学会会員。



田中 聡(正会員)

昭和60年慶應義塾大学理工学部管理工学科卒業。昭和62年日本電信電話(株)入社。オブジェクト指向交換プログラム、通信網用分散処理の研究開発に従事。同社未来ねっと研究所主任研究員。電子情報通信学会会員。



須田 達也

昭和52年京都大学工学部数理工学科卒業。昭和57年同大学大学院博士課程修了。博士号取得。コロンビア大学研究員。カリフォルニア大学アーバイン校研究員を経て、現在カリフォルニア大学アーバイン校情報科学科教授。平成8年より平成11年までNSF Networking Research Program ディレクタ。平成9年より平成11年までIEEE U.S. Society Relations ディレクタなどを歴任。現在、NTT リサーチプロフェッサ、IEEE/ACM Transaction on Networking, IEEE Transaction on Communications 各エディタ。コンピュータ通信、高速ネットワーク、マルチメディアシステム、分散システム、オブジェクト指向通信システム、性能解析に関する研究などに従事。工学博士。IEEE、ACM 各会員。



青山 友紀(正会員)

昭和42年東京大学工学部電子工学科卒業。昭和44年東京大学大学院電気工学科修士課程修了。同年日本電信電話公社(現NTT)入社。以来、電気通信研究所において伝送システム、デジタル信号処理、広帯域ネットワークなどの研究に従事。平成9年東京大学に転じ、現在同大学院情報理工学系研究科電子情報学専攻教授。工学博士。IEEE Fellow。現在本会総務理事。超高速フォトニックネットワーク開発推進協議会会長。デジタルシネマコンソーシアム会長。