

サービス属性を用いたサービス合成方式

今田美幸† 片山 穰† 松尾真人† 須田達也† ‡

† NTT 未来ねっと研究所

‡ University of California, Irvine

〒180-8585 東京都武蔵野市緑町 3-9-11

TEL:0422-59-2546 FAX:0422-59-4810

imada@ma.onlab.ntt.co.jp

概要 我々は、Ja-Net(Jack-in-the-Net)と呼ぶ、環境に適應、進化できるネットワークサービスの研究を行っている。Ja-Net では、環境中に存在する **Cyber Entity (CE)**と呼ぶ自律性と移動性を持つエージェントが、相互に通信しあうことで処理を実行する。単一のサービスプロバイダが生成した CE は、機能やサービス内容が限られているため、CE 単体では個別化するエンドユーザのサービス要求を満たせない場合が多い。そこで、複数の CE を用いて、エンドユーザのサービス選択の幅を広げることが必要になる。

Ja-Net では、CE の種別や数が時々刻々と変化するシステムを前提としている。このため、エンドユーザが利用できるサービスは、エンドユーザの位置や時間によって異なる。

本稿では、CE のサービス内容を表すキーワードで構成されるサービス属性を用いて、複数の CE を自動的に組み合わせてサービスを生成する方式について提案し、個別化するエンドユーザのサービス要求への適應性の観点から高アシュアランスが実現できることを述べる。

キーワード Cyber Entity, エージェント, サービス合成, サービス属性, 適應性, Ja-Net

1. はじめに

現在、インターネットを利用してさまざまなサービスが提供されている。中でも、端末の小型化、高機能化により、多くの人々が手軽にネットワークサービスを利用できるようになってきた。Web サービスのようなネットワーク上で展開されるサービスは、コンテンツやサービス種別が膨大かつ多様化し、追加や変更が頻繁に行なわれる。このような状況で、エンドユーザが快適にサービスを利

用するためには、ネットワーク上でサービスを提供する異なるサービスプロバイダ間の柔軟な連携が重要になってくる。これを実現するために、リアルタイム性やフォールトトレランス性といったような異種性、および状況が変化しても追従できる適應性の 2 つを併せもつ高アシュアランス性[1]の実現技術が不可欠になる。

Web をベースにした e-business 展開のアーキテクチャとして、サービス指向アーキテクチャ(Service-oriented Architecture : SOA)がある。SOA のコンポーネントとしては、サービスプロバイダ (SP), サービスブローカ(SB), サービスリクエスト(SR)がある。SP は、サービスの内容を公開し、利用者を求める。SB は、SP を分野別に分類し、検索サービスを提供する。SR は、SB を利用して SP が提供するサービスを検索し、利用する。

一方我々は、将来のネットワークが人間やコンピュータ、家電や自動車など多種多様なデバイスや装置をノードとしてつなぐユニヴァーサル・ネットワーク(Universal Network)へと変化・発展していくと考えている。このようなユニヴァーサル・ネットワークでは、SP の数が膨大で多種多様となる上、サービス提供時間が SP 毎に異なる。このため、SR がサービスを利用する際、SR の位置やタイミングによって利用可能なサービスが異なる。このような状況下において、SOA の SB のようなサービス内容やインタフェース条件を固定的なサーバで集中管理することは困難になるものと思われる。そこで我々は、SR の近辺のローカルエリアに存在するサービスコンポーネントを用いて、個人あるいは大衆の SR の要求に適應できる有効なサービスを創発、進化、自然淘汰するために、生物界をモデルとしたアーキテクチャ Ja-Net を提案し[2][3][4][5][6]、現在検討を進めている。

本稿では、Ja-Net における高アシュアランス実現方式として、個別化するエンドユーザのサービス要求に対して適應性を持たせる方式について述べる。

2. システム構成とアシュアランス性実現のための課題

2.1 システム構成

A part of Prof. Suda's research conducted at UCI and 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).

図 1 に、システム構成を示す。Ja-Net は、複数のコンピュータノード(以下ノードと略す)から成る。SP や SR は、CE(SP)や CE(SR)を生成し、Ja-Net プラットフォーム上に配置する。

CE(SR)の主な機能は、SR とのインタフェースを提供することである。CE(SR)は、SR に成り代わって、SR の要求の Ja-Net 上に送信したり、そのリクエストに対して応答してきた CE(SP)のサービスを SR の嗜好でフィルタリングして、SR に渡す。

一方、CE(SP) の主な機能は、CE(SR)のサービスリクエストに対し、自身が提供するサービスを CE(SR)を介して提案したり、必要に応じて他の CE(SP)と共同でサービスを実行する。

CE(SP)は、Ja-Net プラットフォーム上で、複製生成、消滅などを行なう。このポリシは、エネルギーによって規定する。ノードの移動や複製の生成のために必要なエネルギー量は、Ja-Net プラットフォームが規定する。CE(SP)が SR に提供したサービスの評価もエネルギーに換算する。評価の程度に応じて、CE(SP)が獲得できるエネルギー量を決める。エネルギーの獲得量の少ない CE(SP)は、Ja-Net プラットフォームによって消滅させられる。獲得量の多い CE(SP)は、エネルギーを使って、複製生成、ノード移動、メッセージ送信範囲の拡大など、Ja-Net プラットフォーム上での活動能力を高めることが可能なる。

CE(SP)が Ja-Net プラットフォーム上で、サービスパートナー検索メッセージを送信する場合にも、エネルギーが必要となる。メッセージの送信範囲は、CE(SP)が保持するエネルギー量によって制限される。

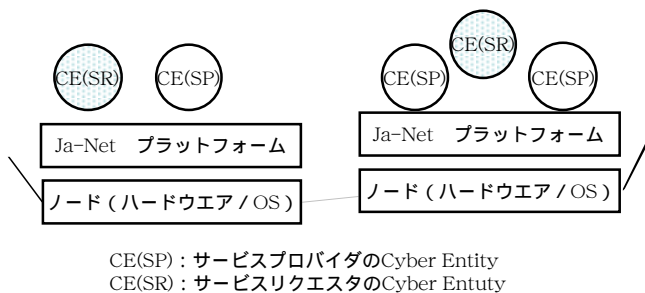


図 1 システム構成

2.2 アシユアランス性実現のための課題

Ja-Net では、1 章で示した SOA の SB が存在せず、SP-SR 間、SP-SP 間のコミュニケーションによって、サービスの創発、発見、利用を実現する。実際のコミュニケーションは、自律的な行動基盤を持つ CE が行なう。

Ja-Net では、CE の数や種類、サービスの寿命を一

括管理することはしない。

このような状況下での SR のサービス利用手順を以下に示す。

- (1)SR(個人や大衆)の位置、嗜好、リクエスト内容を把握する。
- (2)SR(個人や大衆)の位置、嗜好、リクエスト内容にあったサービスを提供する SP を検索する。
- (3)該当する SP が、SR が利用できる端末機器を介して、サービスを提供する。

この一連の動作を高アシユアランスに実現するためには、SR が要求するサービスリクエストと SP が提供するサービスとの迅速な連携が必要となる。このためには、SR の位置情報、サービスリクエスト内容に対し、そのサービスが提供できる SP をリアルタイムにすなわち、物理的な距離の移動時間にあった範囲内で高速に検索しなければならぬ。

方式として以下が考えられる。

- 方式 1 : 短時間で条件にあうサービスを検索する
- 方式 2 : SR がよく利用するサービスを提供する SP のコンテンツ情報を、SR に物理的に近い位置に配置する。
- 方式 3 : ローカルエリア毎に提供できるサービスを決め、エリア単位で提供できるサービスを把握する。
- 方式 4 : ローカルエリアにあるサービスコンポーネントを組み合わせて要求するサービスを生成する。

方式 1 に関しては、Google[12]などの検索エンジンを実現する分野で研究行なわれている。

方式 2 に関しては、SR のサービス利用履歴を元に、有効なサービスの組み合わせを発見し、SP がよく利用されるサービス同士をネットワークの距離の近いノードに、オフラインで違いにネゴシエーションした上で、配置替えし、検索の利便性を向上させる研究も行なわれている[7]。

方式 3 に関しては、SR(人間)の位置移動時間をあらかじめ計算し、SR がサービスが利用提供できるエリアを決め、エリア毎に SR にサービスを提案する研究がある[8][9]。

上記 3 つの方式は、主に、すでに存在しているサービスを目的に併せていかに効率よく検索できるかに関わる研究である。

しかし、Ja-Net のように検索エリアが限られている場合、個別化するサービス要求に十分応えられない場合が

発生する。このため、方式4は重要な検討課題である。つまり、ローカルエリアでの偶然性に頼るのではなく、あらかじめ有効と思われるサービスの組み合わせを作っておいて、提供できるサービスの幅を広げておくことが必要である。

複数の SP がグループとしてサービスを提供すれば、単体では提供できないサービスを提供することができる。例えば、レンタカ予約をする SP の CE を CE(レンタカ予約)と記述する。同様に、ホテル予約、飛行機予約を行なう SP の CE を CE(ホテル予約)、CE(飛行機予約)と記述する。この3種類の異なる CE がグループでサービスを提供する場合、旅行出張予約に使えるサービスができる。

また、構成メンバの SP を変えると、異なるサービスが生成できる。旅行出張予約サービスの構成要素である CE(ホテル予約)は、ケーキを販売する SP の CE(SP)である CE(ケーキ販売)とグループでサービスを提供することで、ケーキバイキングという全く異なるサービスが生成できる。

複数の CE(SP)でサービスを生成することを**サービスの合成**と呼び、合成した結果できるサービスを**合成サービス**と呼ぶ。

本稿では、方式4を実現方法提案し、適応性の観点から高アシュアランス性が実現できることを示す。

3. サービス合成方法の種類

(1) 並列型合成サービス

個々の CE(SP)は、他の CE(SP)の実行結果を利用することなく、並列にサービスを実行する。実行結果は、あらかじめ定めた代表 CE(SP)に返送する。メンバ CE(SP)の全ての処理結果が揃ったら、代表 CE(SP)は合成サービスとして CE(SR)を介して SR に提供する。このようなタイプの合成サービスを**並列型合成サービス**と呼ぶ。例えば、旅行出張予約サービスでは、CE(ホテル予約)、CE(レンタカ予約)、CE(飛行機予約)が各々並列に処理を実行できる。全ての CE の予約の処理が完了したら、旅行出張予約サービスとして SR に提供できる。

並列型合成サービスは、インタフェースよりも、個々の CE(SP)が提供するサービスの利用目的やシーン、タイミングといった利用形態が一致し、共に利用されることが重要である。サービス合成の過程では、利用形態が一致している CE(SP)を発見することが重要になる。

並列型合成サービスを構成メンバとなりうる CE(SP)サービスは、単にコンテンツや商品の売買するという特

徴がある。つまり、SR からコンテンツや取り扱う商品に対する購入のリクエストを受けると、必要に応じて CE(SP)生成元の SP に在庫を確認し、在庫があれば CE(SP)から CE(SR)を介して SR に予約 OK のメッセージを送信する。例えば、音楽コンテンツを SR に配信する SP が生成した CE(音楽配信)や CE(ホテル予約)などがそれにあたる。このようなタイプの CE(SP)を、**コンテンツ情報提供タイプ**と呼ぶ。

(2) 直列型合成サービス

サービス生成の際に、順序性があるサービスを**直列型合成サービス**と呼ぶ。直列型合成サービスは、サービスの組み合わせの順序性を規定するための input / output にあたるキーワードの一致条件を繰り返すことで生成する。本キーワードは、CE オブジェクトを実行するための引数ではなく、あくまでも概念的なサービスの組み合わせの順序性を規定しているものである。

直列合成サービスを構成するメンバには、かならず、入力されたコンテンツに加工処理を行なう CE(SP)が含まれる。SR は、コンテンツ情報提供タイプの CE(SP)が提供するコンテンツと加工イメージ情報を、加工処理を行なう CE(SP)に入力すると、必要に応じて CE(SP)生成元の SP に加工処理可能かどうかを確認し、可能であれば SR に加工後のイメージとサービスの予約 OK のメッセージを返送する。加工処理を施す CE(SP)の例としては、バイキングなどの食事のアレンジを行なう CE(食事アレンジ)や、音楽のテンポを変更する SP が生成した CE(音楽加工)などがある。このようなタイプの CE(SP)を、**加工情報提供タイプ**と呼ぶ。

(2) 混合サービス

直列型合成サービスの入力が並列型合成サービスの実行結果となっているような、並列型と直列型の合成サービスが混合している合成サービスを**混合サービス**と呼ぶ。

混合サービスは、並列型と直列型の組み合わせによって生成できるため、本稿では記述を省略する。

4. サービス合成のための手順

1 章で示した方式4を実現するための手順は次のようになる。

(1)サービスの組み合わせパターンの抽出

(2)CE(SP)間の binding 条件の確認

現在は、(1)の処理を人間(SR)が行ない、(2)をシステム

で自動化する研究は行なわれている [10][11]。しかし、(1)をシステムで完全に自動化するような研究、つまり、STONE のサービスグラフ[10]や UDDI の Service execution transaction[11]を自動生成するような研究はあまり行なわれていない。

本稿では、(1)を自動化するための方式について提案する。

5. サービスの組み合わせパターンの抽出

5.1 サービス属性の分類

本稿では、e-business 分野において、異なるサービスを提供する SP 間を動的に結合することを目的としている。

3 章で示した形態の合成サービスは、サービスの利用形態とサービスの組み合わせの順序性に生成の特徴があった。そこで、サービス属性を分類するフレームワークとして 5W1H を利用することとする。

各ブロックには、CE(SP)の生成者がサービスの特徴をキーワードとして記述し、格納する。キーワードを記述する際、Ja-Net で定めた用語の定義などを示すガイドラインにしたがう。これによって、同義語のオントロジーの差異による不一致を防ぐ。もし、特定のブロックに対し、該当するキーワードがない場合、空欄でもよい。

表 1 に、5W1H の各ブロックとその中に格納するサービス属性を説明する。

What に記述するサービス名は、人間がサービスを利用する際、そのサービス内容を理解するために利用する名前である。これは、人間(SR)に、Ja-Net のローカルエリアで提供できるサービスを提案する際に用いる。詳細は 5.5 節で示す。

Who の識別子は、CE(SP)間のメッセージ送信時に利用する。

サービスの組み合わせの順序性は、How ブロックに記述する。つまり、どのようなタイプのコンテンツや商品情報を input すれば output としてどのようなコンテンツや商品を造り出すことができるかといったようなサービス属性レベルでの input/output のキーワードを記す。加工情報提供タイプの CE(SP)の場合、本ブロックにキーワードが入る。つまり、コンテンツ情報提供タイプの CE(SP)には、本ブロックにキーワードがない。

本稿では、利用形態を利用目的、利用シーン、利用場所に分類し、それぞれ、Why、When、Where ブロックとする。

Details ブロックは、5W1H のフレームワークとして

説明できないサービスの詳細情報を示す。コンテンツ情報提供タイプの CE(SP)ならば、提供するコンテンツの中身や商品情報や種類、商品の構成部品や素材情報等のキーワードを列挙する。

合成サービス生成後は、合成サービスとしてのサービス属性を持つ。合成サービスのサービス属性は、構成メンバの CE(SP)全てが保持するとする。

表 1 サービス属性の分類

ブロック名	説明
What	人間(SR)がサービスの内容を理解できるようなサービス名
Who	識別子
How	サービスの組み合わせの順序性
Why	利用形態 (サービスの利用目的)
When	利用形態 (サービスの利用タイミング)
Where	利用形態 (サービスの利用場所)
Details	サービスの特徴をあらゆる詳細情報

5.2 CE(SP)間の処理シーケンス

5.2.1 並列型合成サービス

並列型合成サービスを生成する場合、3 章(1)で示したように、コンテンツ情報提供タイプの CE(SP)間で、利用形態を表す Why、When、Where のブロックのキーワードが一致するか否かで生成可能性を判断する。本タイプの CE(SP)は、How ブロックには input/output のキーワードがない。

よって、サービスの合成条件としては、異なる 2 つの CE(SP)または合成サービス間において、(How ブロックの input/output のキーワードがない)、または(利用形態のいずれかのブロックのキーワードが 1 つ以上一致)する場合、サービス合成可能であるとする。

例として、CE(a)が、サービスパートナ検索メッセージをローカルエリア内に送信する場合の、サービス合成方法の概要を以下に示す。処理フローは、図 2 に示す。

(1) CE(a)は、CE 生成者があらかじめ定めた契機に、サービスパートナ CE(SP)検索メッセージを送信する。但し、CE(a)は、図 3 に示すサービス属性を持つとする。検索メッセージ送信範囲は、エネルギー量によって規定される。送信するメッセージには、ブロックに対応するキーワードを格納する。ここでは、Why、When ブロックのキーワードを送信する。

(2)メッセージを受取ったCE(c)やCE(o)等のCE(SP)は、CE(a)のメッセージ内に記された Why, When ブロックのキーワードと自身のキーワードを比較する。但し、CE(o)は、図3に示すサービス属性を持つとする。

(3)CE(o)は、When ブロックのキーワード p が一致していることを検出する。

(4)CE(o)は、CE(a)に対し、自身のサービス属性とWhenブロックの p が一致していることをメッセージで通知する。

(5)CE(a)は、サービスレベルで合成可能と判断し、CE(SP)間の binding 可能性を確認する。

(6)確認の結果、binding 可能ならば、サービス合成可能と判断し、CE(a)は CE(o)のサービス属性を用いて合成サービスのサービス属性を生成する。図3に合成サービス(Group)のサービス属性を示す。

What ブロックのサービス名としては、利用形態のブロックで一致したキーワード p と両者の What ブロックのキーワードを and 条件で記述する。Who ブロックの合成サービスの識別子 GID=100 を生成する。最後に残りのブロックのキーワードを、ブロック単位に or 条件で格納する。

並列型合成サービスは、サービスとしての起動順序がない。本稿では、CE(a)をグループの代表CEとし、個々のCEの処理結果を取りまとめて、SRに返送するとする。

生成した合成サービスの起動するために、合成サービスのグループ情報を生成する。表2にグループ情報の概要を示す。グループ情報は、合成サービス名、合成サービスの識別子 GID、合成サービスを構成する先頭のCE(SP)の識別子 init. CEID、終端のCE(SP)の識別子 end CEID から成る。

表2 合成サービス p(a, o)のグループ情報

合成サービス名	GID	init. CEID	end CEID
p(a, o)	100	1	2

また、合成サービスを構成するメンバ CE(SP)の2項間のメッセージ接続情報を個々のCE(SP)が生成する。CE(a)が保持する接続情報を表3に示す。これは、GID、自身CE(SP)の前後に接続する from-to にあたるCEの識別子 CEID の情報からなる。利用方法は、5.3節と5.4節で説明する。

表3 CE(a)の接続情報

GID	from CEID	to CEID
100	なし	2

合成サービスのサービス属性とグループ情報は、全て

のグループメンバのCE(SP)が保持する。

(7)CE(a)は CE(o)に対して、合成サービスが生成できたとこと通知する。その際、合成サービスのサービス属性図3と表2のグループ情報を通知する。

(8)CE(o)は、合成サービスのキーワードとグループ情報を格納する。

さらに、(6)の属性を持つCE(o)の接続情報を生成する。表4にCE(o)の接続情報を示す。利用方法は、5.3節と5.4節で説明する。

表4 CE(o)の接続情報

GID	from CEID	to CEID
100	1	なし

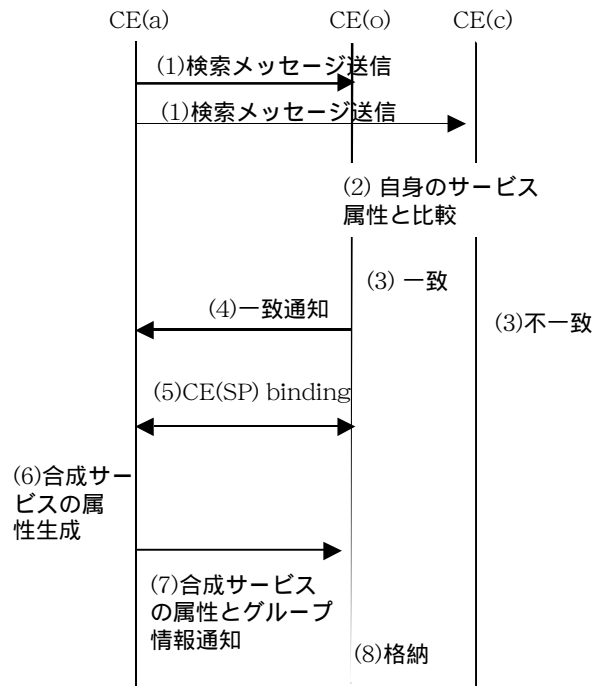
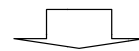


図2 並列型合成サービスの処理フロー

	What	Who	How	Why	When	Where	Details
CE(a)	a	CEID=1	in/out : なし	b	p		xxx yyy
CE(o)	o	CEID=2	in/out : なし		p	q, r	v



	What	Who	How	Why	When	Where	Details
Group	p(a,o)	GID=100	in/out : なし	b	p	q, r	xxx yyy v

図3 CE(a), CE(o)と合成サービス(group)のサービス属性

5.2.2 直列型合成サービス

直列型合成サービスは、3章の(2)で示したように、CE(SP)の How ブロックの input/output のキーワードとコンテンツや商品情報を提供する CE(SP)のキーワードが一致すれば、サービスレベルで合成可能と判断する。

サービスの合成条件としては、2つの異なる CE(SP)または合成サービスにおいて、(一方の How ブロックの input のキーワードと他方の Details ブロックのキーワードが一致)、または(一方の How ブロックの output のキーワードと他方の Details ブロックのキーワードが一致)、または(一方の How ブロックの output のキーワードと他方の How ブロックの input のキーワードが一致)した場合、サービス合成可能と判断する。

例として、CE()が、サービスパートナ検索メッセージをローカルエリア内に送信する場合の、サービス合成方法の概要を以下に示す。

(1)CE()は、CE 生成者があらかじめ定めた契機に検索メッセージを送信する。送信する検索メッセージには、サービスパートナ CE に求めるブロック名と対応するキーワードを格納する。ここでは、How ブロックの input/output のキーワードを送信する。CE()は、図4に示すサービス属性を保持しているとする。

(2)メッセージを受取った CE()や CE()等の CE(SP)は、それぞれ、CE()のメッセージ内に記された How ブロックのキーワードと自身の How または Details ブロックのキーワードを比較する。ここで、CE()は、図4に示すサービス属性を保持しているとする。

(3)CE()は、CE()の output のキーワードと自身の Details ブロックの w が一致していることを検出する。

(4)CE()は、CE()に対し、自身の Details ブロックと output が w で一致していることをメッセージで通知する。

(5)-CE()は、サービスレベルで合成可能と判断し、CE(SP) binding が可能か確認する。

(6)その結果、binding 可能ならばサービス合成可能と判断し、CE()は合成サービスのサービス属性を生成する。図4に合成サービス(group)のサービス属性を示す。

What ブロックのサービス名には、両者の What ブロックのキーワードを and 条件で記述する。Who ブロックの合成サービスの識別子 GID=200 を生成する。How ブロックの output のキーワードはなしとし、Details ブロックに w を記述する。input のキーワードおよびその他のキーワードは or 条件で格納する。最後に残りのプロ

ックのキーワードを、ブロック単位に or 条件で格納する。

合成サービスのサービス属性とグループ情報は、全てのグループメンバの CE(SP)が保持する。

生成した合成サービスの起動するために、合成サービスのグループ情報を生成する。表5にグループ情報の概要を示す。また、CE()が保持する接続情報を表6に示す。

表5 合成サービス - のグループ情報

合成サービス名	GID	init. CEID	end CEID
-	200	10	11

表6 CE()の接続情報

GID	from CEID	to CEID
200	なし	11

利用方法は、5.3節と5.4節で説明する。

合成サービスのサービス属性とグループ情報は、全てのグループメンバの CE(SP)が保持する。

(7)CE()は CE()に対して、合成サービスが生成できたこと通知する。その際、図4の合成サービスのサービス属性と表5のグループ情報を通知する。

(8)CE()は、合成サービスのサービス属性とグループ情報を格納する。

次に、CE()の接続情報を生成する。概要を表7に示す。

表7 CE()の接続情報

GID	from CEID	to CEID
200	10	なし

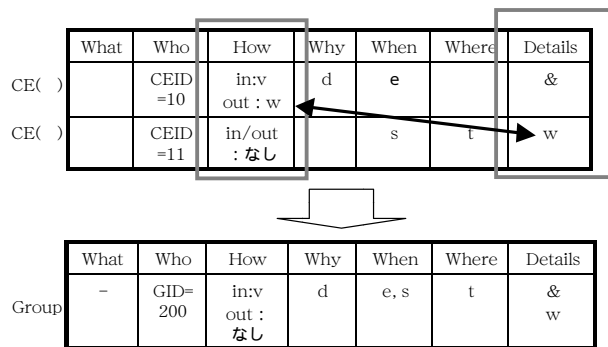


図4 CE(), CE()と合成サービス(group)のサービス属性

5.3 合成サービスの実行

合成サービスを起動する際は、表2や表5のグループ

情報を使う。

人間(SR)がサービス要求を送信する場合、合成サービス名で指定する。合成サービス名を受け取った個々のCE(SP)はグループ情報の init. CEID を見る。もし、それが自身の CEID ならば、合成サービスの先頭なので、合成サービスを起動する。init. CEID が自 CEID でなければ、init. CEID に記載の CE(SP)に起動メッセージを送信する。

次に、先頭の CE(SP)で処理を実行したら、表 3 や表 6 接続情報の to CEID の記載の CE(SP)に、メッセージを送信する。表 4 や表 7 のように、to CEID に CE(SP)のデータがない場合、合成サービスの終端とみなし、サービス要求した SR に対し、CE(SR)を介してサービスを提供する。この時点でサービス実行が終了したとみなす。

5.4 合成サービスの拡張

合成サービスの拡張は、構成する CE(SP)のグループを 1 つの CE(SP)とみなして、サービスパートナーを検索し、条件にあうパートナーがいた場合、合成サービスの拡張が可能と判断する。

例として、合成サービス - と CE(o) 間での拡張を示す。

合成サービス - は、加工情報提供タイプなので、拡張は直列型合成サービスと同様に行なう。5.2.2 で示したように、合成サービス - と CE(o) 間でサービス属性の比較を行う。図 5 にサービス属性を示す。

合成サービス - の input のキーワード v と CE(o) の Details のキーワード v が一致したとする。よって、サービスレベルで合成可能と判断し、CE(SP) binding の可能性を確認し、問題なければ、サービス合成可能と判断し、合成サービス o- - を生成する。合成サービスのサービス属性は、CE(o) が生成する。

次に、グループ情報と接続情報を生成する。図 6 に概要を示す。

表 5 の合成サービス - のグループ情報において、合成サービスを構成する CE(SP)チェーンの先頭か終端のいずれに CE(o)を接続すべきかについては CE()によって決まる。ここでは、CE()の入力として接続とする。合成サービス o- - のグループ情報は CE(o)が生成する。

CE(o)は、合成サービスの先頭なので、図 5 の合成サービスの属性とグループ情報を合成サービス - に通知する。この時、CE(o)は接続情報を生成する。

合成サービス - の先頭 CE()は、CE(o)から送信された合成サービスの属性とグループ情報を格納し、新

規に GID=300 の接続情報を生成する。次に、表 6 を使って、CE()に、CE(o)から送信された合成サービスの属性とグループ情報を送信する。受信した CE()は、これらの情報を格納し、GID=300 の接続情報を生成する。

これで、合成サービス o- - への拡張が完了する。

	What	Who	How	Why	When	Where	Details
Group	-	GID=200	in:v out:なし	d	e, s	t	& w
CE(o)	o	CEID=2	in/out:なし		p	q, r	v

	What	Who	How	Why	When	Where	Details
Group	o-	GID=300	In/out:なし	d	e, s, p	t, q, r	&, w, v

図5 合成サービス -, CE(o), 合成サービス o- - のサービス属性

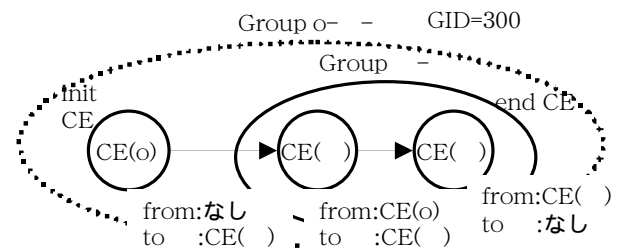


図6 グループ情報と接続情報

5.5 合成サービスの妥当性の判断

本提案方式は、サービス属性を用いて Ja-Net が自動的にサービスを合成する。よって、生成した合成サービスが SR にとって有益であるかを判断するためのロジックが必要となる。

この判断には、SR のサービス要求を契機に、What ブロックに格納してある合成サービスの名前で提案する。本サービス名は、人間(SR)が合成サービスの内容を理解できるので、そこでサービスを選択すれば有効であったと判断できる。選択されなければ、無効なサービスであったと判断し、合成サービスを削除する。

6. 考察

本提案方式は、サービスをシステム自動で合成する際、人間を介すことなく、無効な合成サービスを少なく生成するために用いる。これは、サービスの利用形態やサービスの組み合わせの順序性に着目して合成を実施したた

めである。

本方式を既に研究されている[10][11]と組み合わせることで、サービス合成の完全自動化が実現できる。

本提案方式は、Ja-Net に対するサービスリクエストとは非同期に実行するため、広域なサービス検索をしなくとも、ローカルエリアにおける SR のサービスに対する選択の幅が広がる。

本方式は、曖昧なサービスリクエストに対して、利用可能なサービスをリアルタイムに提案できるため、曖昧なサービスリクエストを明確なサービスリクエストに変えるための支援ができる。

つまり、ローカルエリアにおける利用可能なサービスの多様性の向上、リアルタイムなサービス提供が可能となるため、アシュアランス性向上につながる。

本提案方式は、サービスのキーワードを元に新しいサービスを生成することを目的としている。よって、適応範囲として、ネットワークに複数の SP が参入するような e-business に適している。また、新しいサービスや商品、システムを考案する分野にもてきしている。しかし、システムを制御するような分野においては、さらなる改良が必要と考える。

7. まとめ

多種多様で莫大な数の CE(SP)が存在する Ja-Net 環境において、ローカルエリアのサービス検索だけで、SR の個別化する要求に対応するためのサービス合成の自動化方式を提案した。これは、CE(SP)のサービス属性を 5W1H のフレームワークにしたがってキーワードを分類し、分類したブロック単位にサービスが合成可能か否かを判断する。もし、サービス属性レベルで合成可能と判断したら、CE(SP) binding が可能か判断する。

Ja-Net では SR のサービスリクエストとは非同期に、CE(SP)が自律的に合成タイミングをきめる。よって、SR からリクエストが送信された時には、既にできている合成サービスを提案できる。この際、人間が理解できるサービス名が付いているため、SR はそれが自身にとって有効か否かが判断し易い。

本提案方式は、ローカルエリアにおける利用可能なサービスの柔軟な適応やリアルタイムなサービスの提案が可能のため、アシュアランス性を向上できる。

現在は、基本設計とサービスアプリケーションに対応した合成サービス生成ロジックの設計を進めている。

今後は、混合サービスの詳細検討や CE(SP) binding についてさらに詳細設計を行なう。

謝辞

本研究を進めるにあたり、有益な助言や討論をして頂いた東京工業大学 森欣司教授、および同研究室の皆様に深く感謝致します。

参考文献

- [1]K. Mori: "Autonomous Fading and navigation for Information Allocation and Search under Evolving Service System", Proc. of Asia Pacific Symposium on Information and Telecommunication technologies (APSITT '99), pp.326-330, 2000.
- [2] M. Wang and T. Suda, "The Bio-Networking Architecture: A Biologically Inspired Approach to the Design of Scalable, Adaptive, and Survivable/Available Network Applications," Proceedings of the IEEE SAINT Symposium, 2001.
- [3] T. Suda, T. Itao, T. Nakamura and M. Matsuo, "Adaptive Networking Architecture for Service Emergence," Trans. Inst. Electronics Communication Engineers of Japan (IECEJ), Vol. J84,-B, No.3, pp.310-320, Invited Paper, 2001.
- [4] T. Itao, T. Suda, T. Nakamura and M. Matsuo, "Adaptive Networking Architecture for Service Emergence," in the Workshop on Highly Distributed Systems in the IEEE SAINT Symposium, 2001.
- [5] N. Miyamoto, "An Equilibrium Model of a Self-organizing Network Architecture", Master Thesis, Kyoto University, 2001.
- [6] S. Song and T. Suda, "Security on Energy Level in the Bio-Networking Architecture," Proceedings of the 3rd International Conference on Advanced Communications Technology, 2001.
- [7]H. Arfaoui and K. Mori: "Autonomous Integration of information Services in Heterogeneous FIF System", Proc. 2000 international Workshop on Autonomous Decentralized System (IWADS 2000), pp.40-45, Sep. 21-23, 2000.
- [8]大野敬典, 森欣司: "サービス性向上のための利用場所と時間を考慮した情報配布技術", 2001年電子情報通信学会講演論文集, D-9-22.
- [9]ソーサワ・センサティット, 鈴木泰山, 森欣司: "特定エリア不特定多数の要求を反映した動的サービス情報選択技術", 2001年電子情報通信学会講演論文集, D-9-23.
- [10]<http://www.uddi.org>
- [11]南 正輝, 森川博之, 青山友之: "ネットワークサービスシンセサイザのデザイン", 信学技報, SSE2000-236, IN2000-193, March 2001.
- [12]<http://www.google.com/>