

# LAQoS：多階層構造による適応的 QoS モデル

滝 沢 泰 久<sup>†</sup> 桐 本 直 樹<sup>†</sup> 倉 林 則 之<sup>††</sup>  
張 兵<sup>†</sup> 山 崎 達 也<sup>†††</sup>

インターネットの普及とマイクロプロセッサの急速な進歩にともない、分散型/移動型マルチメディアシステムが多く出現している。このようなシステムは、複数の任意のサービスが混在し、不均質に変動するリソース環境において、多種多様で嗜好性をともなうユーザ要求を満たすことが求められる。この要求を満たす分散/移動型マルチメディアシステムを実現するため、ネットワークにおけるサービス品質の保証 (Quality of Service: QoS) に加えて、ネットワークのエンドシステムにおける QoS メカニズムの研究が行われている。しかし、エンドシステムにおいては、QoS メカニズムははまだレイヤ内に閉じており、また、特定の単一サービスを想定した環境において、QoS を提供するにとどまっている。そのため、我々は、複数のサービスが混在し、事前に予測が困難なエンドシステム環境において、個々機能の制約を考慮しつつ、システム全体として合目的に向かう理論的処理モデルが必要と考え、Layered Adaptive QoS (LAQoS) モデルを考案した。本論文では、エンドシステム内の QoS 統合モデルとして、レイヤ間の相関を積極的に取り入れた LAQoS モデルを提案する。さらに、LAQoS モデルを数理モデル化し、その数理モデルから LAQoS モデルの特性を示す。

## LAQoS: An Adaptive QoS Model Using Layered Structures

YASUHISA TAKIZAWA,<sup>†</sup> NAOKI KIRIMOTO,<sup>†</sup>  
NORIYUKI KURABAYASHI,<sup>††</sup> BING ZHANG<sup>†</sup> and TATSUYA YAMAZAKI<sup>†††</sup>

Distributed multimedia systems and mobile multimedia systems are developed and began to be used widely as spread of Internet and improvement of microprocessor. Such systems are required to meet multifarious and inconstant user's requests on dynamically fluctuating computation resources environment. In order to realize the distributed/mobile multimedia systems which meet this requirements, it is need that the QoS mechanisms for every layer are integrated on environment where two or more services are intermingled. In this paper, we propose Layered Adaptive QoS (LAQoS) model which constructs multi layers based on Extended Braitenberg architecture and Subsumption architecture in order to integrate any QoS mechanisms, and present a basic characteristics for proposed model.

### 1. はじめに

広帯域のインターネットの普及とマイクロプロセッサの急速な進歩により、分散型/移動型マルチメディアシステムが多く出現している。このようなシステムの処理環境においては、ネットワークリソースとコンピューティングリソースは一定でなく、不均質に変動をする。一方、ユーザのシステムへの要求は、動画や音声などのメディア処理の要求に見られるように、多

種多様で嗜好性があり、複数のサービスを必要とする。また、同一ユーザにおいても、その使用目的により、システムへの要求は大きく異なってくる。さらに、マルチユーザ環境ではユーザ間に相関 (協調や抑制) が発生する。したがって、マルチメディアシステムは、不均質に変動するコンピューティングリソース環境において、ユーザ間の相関を考慮しつつ、多種多様に変動するユーザ要求を満たすこと (以降、マルチメディア通信コーディネーション) が求められる。このような要求を満たす分散/移動型マルチメディアシステムを実現するため、ネットワーク QoS に加えて、エンドシステムのサービスにおいて、QoS を提供するメカニズムの研究<sup>14),15),17),18)</sup>が多く行われている。しかし、エンドシステムにおいては、この QoS メカニズムははまだレイヤ内に閉じており、また、特定の単一

<sup>†</sup> 株式会社 ATR 適応コミュニケーション研究所  
ATR Adaptive Communications Research Laboratories

<sup>††</sup> 富士ゼロックス株式会社  
Fuji Xerox Co., Ltd.

<sup>†††</sup> 独立行政法人通信総合研究所  
Communications Research Laboratory, Independent  
Administrative Institution

サービスを想定した環境において、QoSを提供するにとどまっている。したがって、我々はマルチメディア通信コーディネーションを実現するためには、ネットワーク QoS 機能に加えて、次の 2 つの制御メカニズムを有する QoS 機能がエンドシステムにおいて必要であると考えられる。

- 各機能レイヤにおいて、複数の任意の QoS サービスが混在し、相互にサービスが競合する環境に適時適応する QoS サービスの適応メカニズム。
- 各機能レイヤの QoS サービスの適応メカニズムをサービス利用側と提供側の相関でレイヤ間で接続し、システム全体として合目的に向かう統合制御メカニズム。

我々は、エンドシステムにおけるこのような制御メカニズムとして、Layered Adaptive QoS (以降、LAQoS) モデル<sup>11)</sup>を考案した。LAQoS モデルでは、サービスの振舞いはサービスの構成要素の相関により形成されると考え、拡張ブラインベルグアーキテクチャ<sup>7)</sup>(以降、EBA)に基づき、各機能レイヤにおける QoS サービスの適応メカニズムとレイヤ間の相関をモデル化する。さらに、サブサブシステムアーキテクチャ<sup>6)</sup>に基づき、上位レイヤサービスが下位レイヤサービスの適応メカニズムを制御するメカニズムをモデル化する。この 2 点により、LAQoS モデルは、複数レイヤ/複数サービスから構成されるシステムを 1 つの適応処理システムとしてモデル化する。

本論文では、エンドシステム内の QoS 統合モデルとして、レイヤ内/レイヤ間のサービスの相関による相互作用に基づいた新たな処理モデルである LAQoS モデルを提案する。さらに、LAQoS モデルを数理モデル化し、その数理モデルから LAQoS モデルの特性を示す。

以下、2 章で、LAQoS モデルの適応メカニズムモデルと統合メカニズムモデルを示す。次に、3 章で、LAQoS モデルを数理モデル化し、4 章では、その数理モデルを用いて LAQoS モデルの基本的特性を示す。

## 2. LAQoS の適応メカニズムモデルと統合メカニズムモデル

### 2.1 既存の QoS アーキテクチャにおけるレイヤ構造

代表的な QoS アーキテクチャとして、RM-ODP<sup>1)</sup>、QoS-A<sup>2)</sup>、Quartz<sup>3)</sup>、CORBA<sup>4)</sup>などがある。これらのアーキテクチャは、ネットワークシステムにおけるアーキテクチャであるが、一部エンドシステムの機能も含む。いずれのアーキテクチャもレイヤ構造を構成

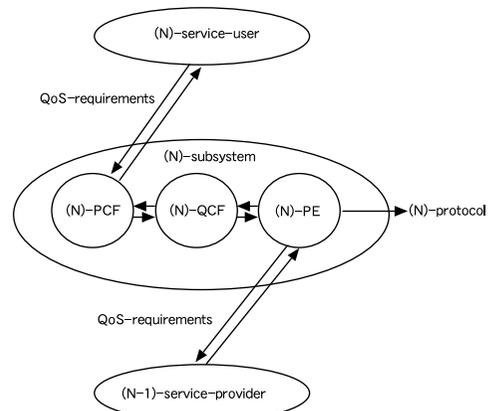


図 1 OSI Negotiation メカニズム  
Fig. 1 OSI Negotiation mechanism.

する点で共通する。このレイヤ構造において、各レイヤにおけるサービス(ゴール)を実現するため、サービス要求をサブサービス(サブゴール)に分割し、サブサービスを提供する下位レイヤ(サブシステム)へサービス要求を行う。要求を受けた下位レイヤは、同様に、サービス要求を分割し、さらに下位のレイヤのサービスを呼び出す。このように上位から下位へ分割されながらサービス要求が行われ、そのサービス実施結果が下位から上位へと統合され、応答されることによりサービスが実現される(例として、RM-ODPにおける Negotiation メカニズムを図 1 に示す)。このようにして、各レイヤの QoS メカニズムが首尾一貫して統合するメカニズムを提供する。すなわち、これらの QoS アーキテクチャはゴール指向システムを構成する。また、レイヤ間の透過性を実現するため、各レイヤ間の依存関係は可能な限り排除される。

一方、想定する処理環境は分散型/移動型マルチメディア処理環境である。このような処理環境は、開放型システムである。また前述のように、ユーザにおける多種多様な嗜好性や移動などともなうサービス品質/コンピューティングリソースの変動などが予想される。このような変動をとともう処理環境では、事前にすべての状態を予測することは不可能であり、未知の状態が発生する。しかし、前述のゴール指向のアーキテクチャはその目標(ゴール)状態とそれに対応する振舞いを詳細に定義する必要があるため、予測できない未知の状態において有効に機能しない。仮に、すべての状態を予測できたとしても、振舞いの選択のためのコストが非常に大きくなる。さらに、開放型システムでは、ユーザの嗜好性やリソース利用状況は変動することから、これらが一致しない場合が発生する。こ

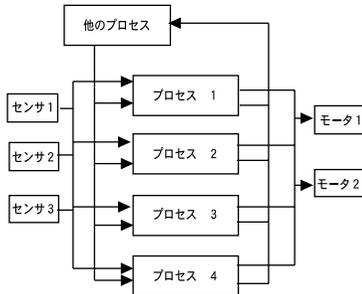


図2 EBAアーキテクチャ  
Fig.2 EBA architecture.

これは、各サービスが異なる価値により QoS を提供するため、レイヤ内サービスにおける競合およびレイヤ間のサービスにおける不整合が生じるためである。この競合や不整合を解消するためには、サービス間で相互に調整するフィードバックが必要であるが、サービス間の透過性を重視するアーキテクチャにおいて、サービス間の相互作用を適用することは不合理である。

## 2.2 LAQoS の適応メカニズムモデル

ゴール指向のメカニズムでは未知の状態に対応できないことから、LAQoS モデルでは、状態から直接的にサービスの振舞いを決定するのではなく、知覚された状態においてサービスを構成する各要素の相関によりサービスの振舞いが形成されるものとして考える。また、従来の階層型アーキテクチャはサービス間の相互作用の適用に向かないことから、相互作用によるアーキテクチャを用いる。この2点から、LAQoS モデルでは身体性認知科学のEBAを用いる。このアーキテクチャはブライテンベルグピークル<sup>8)</sup>の考えを、より一般的なエージェントに拡張し、相互作用メカニズムを基に未知の状態に適応するアーキテクチャである。EBAでは、図2に示すように、環境変動に対応した複数の入力、その入力に応じて内部状態を変化させ、その状態を出力する複数のプロセス、プロセスの出力に応じて環境に作用する複数の出力から構成される。また、プロセス間においてもその出力により相互に作用し合う。EBAでは、状態に応じた振舞いを定義するのではなく、環境やプロセス間の相互作用の構成を定義し、この相互作用の結果として振舞いが形成される。したがって、環境がどのように変動しても、それに伴って振舞いが形成されることとなる。

LAQoS モデルでは、EBAを図3に示されるユーザ、アプリケーション、システムの3つ QoS レイヤ<sup>10)</sup>の個々のサービスごとに適用する。すなわち、サービスはサービス内部の複数のプロセス、外部環境からサービスへの入力およびサービスから外部環境への出力に

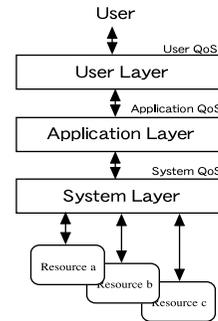


図3 QoSレイヤモデル  
Fig.3 QoS layered model.

より構成される。LAQoS モデルでは、サービス内部のプロセスをサービスを構成する要素として考え、それをエンティティと呼び、次のように定義する。

- 外部(環境および他のエンティティ)からの入力を受ける。
- 外部入力に応じて内部状態を変化させる。
- 内部状態に応じて外部(環境や他のエンティティ)に出力を行う。
- エンティティの出力の値は、エンティティ間の相対的な重要度を表す。

さらに、レイヤ間の各サービスは、サービスの提供側と利用側との関係において相互に影響する点を積極的に取り入れ、レイヤ間の各サービスにおいてもEBAの相互作用メカニズムを適用し、多階層全体で1つの適応システムとしてモデル化する。以下、各レイヤにおけるサービスの適応メカニズムとレイヤ間におけるサービスの相互作用メカニズムのモデルを示す。

### 2.2.1 システム(リソース管理)レイヤ

システム(リソース管理)レイヤは、コンピューティングリソースごとにサービスが提供される。たとえば、CPUであればスケジューリングであり、メモリであれば仮想記憶システムであり、ディスクであればファイルシステムがそれぞれサービスに相当する。システムサービスは、アプリケーションレイヤからのQoS要求と自らのリソースの管理制約に基づき、サービスを提供する(図4参照)。したがって、システムレイヤでは、EBAに基づいて、個々のサービスの適応メカニズムを次のようにモデル化する(図5参照)。

- 環境から入力は管理するリソースの使用状況とアプリケーションからのシステム QoS 要求とする。
- エンティティはリソースを利用するアプリケーションに対応させる。
- エンティティの出力はサービスの振舞いを形成し、リソースと QoS 要求を行ったアプリケーション

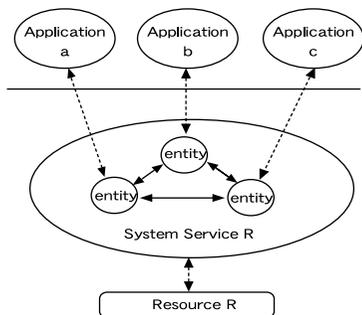


図4 システムサービスの構成  
Fig. 4 Construction of system service.

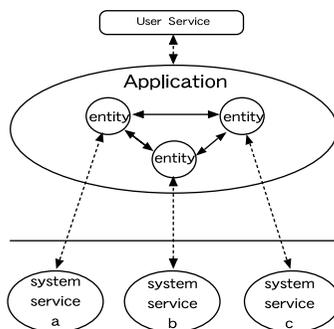


図6 アプリケーションサービスの構成  
Fig. 6 Construction of application service.

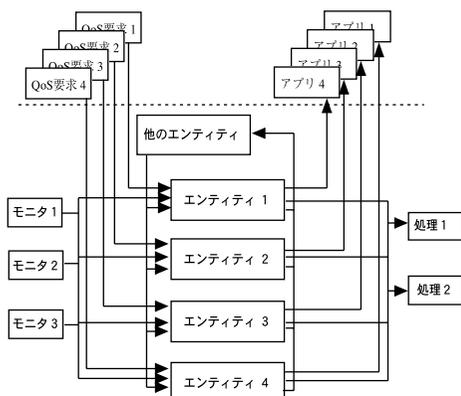


図5 システムサービスの適応メカニズムモデル  
Fig. 5 Adaptation mechanism model for system service.

に作用する。

すなわち、システムレイヤの個々のサービスの適応メカニズムは、QoS 要求およびリソースの利用状況を入力とし、各 QoS 要求に対応するエンティティ(アプリケーション)間の相互作用により行われる。この相互作用の結果としてエンティティは出力を行い、それがリソース内におけるエンティティの重要度、すなわち、アプリケーションの重要度を生成する。この重要度は、当該リソースを利用するアプリケーションレイヤのサービスにフィードバックされる。また、各エンティティの出力(重要度)からリソースの処理が形成される(いくつかの処理が合成されたり、または順序付けされたりする)。

例として、スケジューリングを考える。アプリケーションからの QoS 要求、CPU の利用状況、処理遅延時間、ブロッキング時間などが環境から作用として入力され、各アプリケーションに対応するエンティティの内部状態を更新する。さらにその内部状態に応じて、エンティティは出力を行う。この出力値をタスクの優先度として実行順を決定し(サービスの振り分け)、ス

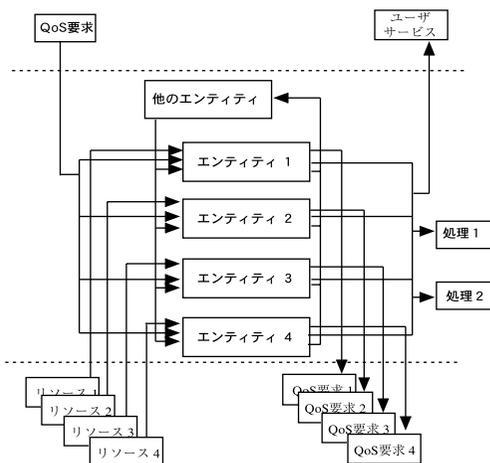


図7 アプリケーションレイヤの適応メカニズムモデル  
Fig. 7 Adaptation mechanism model for application layer.

ケジューリングとしての QoS を提供する。このような適応メカニズムは各サービス(スケジューリング、仮想記憶システムなど)ごとに独立して動作し、その構成(エンティティ間の相関やエンティティと環境との相関)はリソースの管理方式により決められる。

### 2.2.2 アプリケーションレイヤ

アプリケーションレイヤは、複数のアプリケーションサービスから構成される。アプリケーションサービスは、ユーザレイヤから要求されたアプリケーション QoS に基づき、いくつかのシステムサービスを用いて、サービスを提供する(図6参照)。したがって、EBA に基づいて次のようにアプリケーションサービスの適応メカニズムをモデル化する(図7参照)。

- 環境からの入力、ユーザレイヤからの QoS 要求と利用するシステムサービスからのフィードバックとする。
- エンティティはアプリケーションで利用するシス

テムサービスのリソースとする。

- エンティティの出力は対応するシステムレイヤのサービスへの QoS 要求および QoS を提供するユーザレイヤのサービスへ作用する。

すなわち、アプリケーションレイヤの個々のサービスの適応メカニズムは、ユーザサービスから当該アプリケーションへの QoS 要求および当該アプリケーションが利用する (QoS 要求先) システムサービスからのフィードバックを入力とし、利用するシステムサービスに対応するエンティティ (リソース) 間の相互作用により行われる。この相互作用の結果としてエンティティは出力を行い、それがアプリケーション内におけるエンティティの重要度、すなわち、リソースの重要度を生成する。この重要度は、ユーザサービスへ QoS 要求のフィードバックとして、および利用するシステムサービスへの QoS 要求として作用する。また、各エンティティの出力 (重要度) からリソース量に応じたアプリケーションの処理が形成される。

例として、動画再生を行うアプリケーションサービスを考える。動画処理において、各システムサービスの出力により、CPU リソースの出力が低下し、メモリリソースの出力が上昇するなら、フレーム更新レートが抑えられ画像サイズを大きくなるような振舞いを形成する。このような動作により環境に適応する。この適応メカニズムはアプリケーションサービスごとに独立に動作し、その構成 (エンティティ間の相関やエンティティと環境との相関) はアプリケーションの特性により決まる。

### 2.2.3 ユーザレイヤ

ユーザレイヤは、複数のアプリケーションを用いてユーザサービスを提供するユーザエージェントやユーザインタフェースなどの機能レイヤである。ユーザサービスは、ユーザ QoS に基づき、アプリケーションの実行を制御する (図 8 参照)。したがって、EBA に基づいて次のようにユーザサービスの適応メカニズムをモデル化する (図 9 参照)。

- エンティティはユーザサービスを構成するアプリケーションとする。
- 環境からの入力、ユーザからのユーザ QoS 要求と、サービスを構成する複数のアプリケーションの実行状態に応じたアプリケーションレイヤの該当エンティティの出力とする。
- エンティティの出力は、アプリケーションサービスへの QoS 要求とする。

すなわち、ユーザレイヤの個々のサービスの適応メカニズムは、ユーザから当該サービスへの QoS 要求お

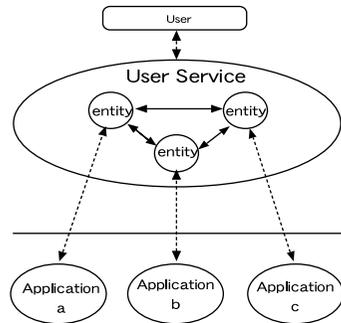


図 8 ユーザサービスの構成  
Fig. 8 Construction of user service.

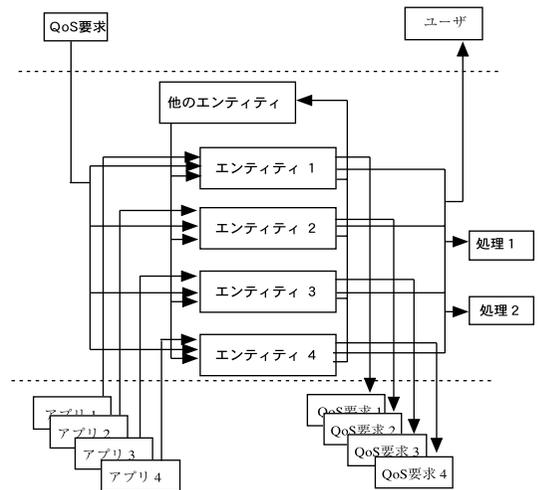


図 9 ユーザレイヤの適応メカニズムモデル  
Fig. 9 Adaptation mechanism model for user layer.

および当該サービスが利用する (QoS 要求先) アプリケーションからのフィードバックを入力とし、当該サービスで利用するアプリケーションに対応するエンティティ間の相互作用により行われる。この相互作用の結果としてエンティティは出力を行い、それがサービス内におけるエンティティの重要度、すなわち、アプリケーションの重要度を生成する。この重要度は、利用するアプリケーションへの QoS 要求として作用する。また、各エンティティの出力 (重要度) から複数のアプリケーションが組み合わせられたり、または順序付けされたりすることにより、ユーザサービスの振舞いが形成される。

例として、複数の動画再生を行うアプリケーションにより構成される多地点のビデオ会議サービスを考える。ビデオ会議サービス外部において CPU 負荷が増大した場合、CPU のシステムサービスは、CPU を利用するアプリケーションに対して均等に CPU リソース

を減ずる出力を行う。しかし、ユーザはある特定の動画再生(たとえば、発言者の動画)に興味が発生した場合、システムサービスが提供する QoS とユーザの嗜好が一致しない。このよう状況において、ユーザが現在の嗜好を反映した QoS 要求(QoS 要求の変更)をユーザサービスに行った場合のシステム全体の動作について述べる。ユーザからの QoS 要求はユーザサービスを構成する特定のアプリケーションの重要度を高める入力として、ユーザサービス内のエンティティ間の相互作用に作用する。その結果、各エンティティの出力が生成され、特定のアプリケーションに対応するエンティティの出力が高くなり、相対的に他のエンティティの出力は低下する。この各出力はそれぞれエンティティに対応するアプリケーションへの QoS 要求として当該のアプリケーションサービスに作用する。したがって、特定のアプリケーションへの QoS 要求はその重要度を高める入力となり、他のアプリケーションへの QoS 要求はその重要度を低下させる入力となる。この入力により、各アプリケーションは利用するリソースに対応するエンティティの相互作用を行い、その出力を生成する。結果として、特定のアプリケーションにおいて CPU リソースに対応するエンティティの出力が高められ、他のアプリケーションでは当該エンティティの出力は低下する。さらに、これらの出力は、各アプリケーションの利用するシステムサービス(リソース)への QoS 要求となる。したがって、CPU をリソースとするシステムサービスは、特定のアプリケーションからの QoS 入力が大きく、その他は小さい入力環境で、サービスを利用するアプリケーションに対応したエンティティの相互作用を行うこととなる。その結果として、CPU 管理制限内において各アプリケーションの重要度をエンティティの出力として生成する。この出力は CPU リソースの割当てを形成するとともに、QoS 要求のフィードバックとして当該アプリケーションに作用する。各アプリケーションは複数のリソースを利用することから、複数のシステムサービスから QoS 要求のフィードバックを受け取る。このような入力から、各アプリケーションはエンティティ間の相互作用を再び行い、各エンティティの出力を合成してユーザサービスへ QoS 要求のフィードバックとする。フィードバックを受けたユーザサービスは、同様に、これを入力としてエンティティ間の相互作用を行う。その結果として生成されたエンティティの出力は再びアプリケーションへの QoS 要求となる。このような処理を各レイヤ/各サービスで非同期に繰り返す、平衡状態(上記の例では、CPU リソースの制

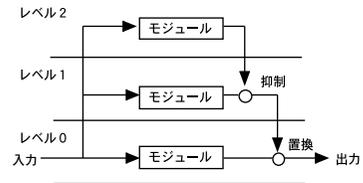


図 10 サブサブジョンアーキテクチャ  
Fig. 10 Subsumption architecture.

約内で特定のアプリケーションに CPU 利用を多く割り当て、他のアプリケーションは相対的に少なく割り当てる状態)へ至る。LAQoS モデルでは、このようなユーザの嗜好、アプリケーションおよびシステムのリソース制限との間で相互作用を繰り返し、システム全体として調整/整合を図る。

我々は、LAQoS の具体的な適用システムとして、複数のユーザにおいて動画再生サービスとチャットサービスを同時に提供するマルチメディアチャットシステム<sup>19)</sup>を検討している。

### 2.3 LAQoS モデルの統合メカニズムモデル

LAQoS では、レイヤ間のサービスにおいて、サービスの提供側と利用側のエンティティが相互に影響(作用)し合う関連を構成する。LAQoS では、このレイヤ間のサービスにおけるエンティティの相互作用を制御するメカニズムが統合メカニズムとなる。この統合メカニズムのモデルとして、サブサブジョンアーキテクチャを用いる。

サブサブジョンアーキテクチャは、EBA と同様に、非同期に振る舞うタスクにより構成される。しかし、EBA はフラットな構造であるのに対し、このアーキテクチャでは、上位レベルのタスクの振舞いが下位レベルのタスクの振舞いを制御(抑制/置換)するための包摂構造を持つ(図 10 参照)。この包摂構造と抑制/置換の制御により、下位エンティティにまったく異なった動作を形成させる。LAQoS では、レイヤ間のサービスにおけるエンティティの相関(提供側と利用側)にこの包摂構造を適用し、EBA による相互作用メカニズムを制御する。すなわち、上位エンティティが下位エンティティの入力を操作し、異なる動作を形成させることにより、環境からの入力が大きく変動した場合におけるレイヤ間の統合メカニズムを提供する。下位エンティティの入力の操作を行うために、上位エンティティが下位エンティティとの包摂構造を知識として持つものとする。この知識は、包摂する下位エンティティ群による適応メカニズムのメタ知識である。

相互作用メカニズムを制御するシステムとして免疫

システムがある．免疫システムにおいて，任意の抗原に適應するために，つねに抗体を生成/削除の繰り返し，抗体のネットワーク構造を変化させている．このメカニズムは，メタダイナミクス<sup>12)</sup>と呼ばれている．LAQoSでは，各エンティティがシステム内の要素に対応していることから，このようなダイナミクスは適用できない．そこで，新しい入力が発生した場合，複数のエンティティの相互作用のメカニズムを変化させるため，メタダイナミクスに相当する入力を，上記のメタ知識を用いて包摂する下位エンティティに行うこととする．この入力をメタ入力と呼ぶ．メタ入力は，下位エンティティに作用し，サブサブジョンアーキテクチャにおける置換/抑制に相当するように，EBAに基づく入力とはまったく異なる性質の入力を下位エンティティに与えることとする．

2.3.1 ユーザサービスの包摂構造

ユーザサービスは，サービスを構成するアプリケーションに対応したエンティティ(ユーザエンティティ)を包摂する(図8参照)．ユーザサービスは，包摂するユーザエンティティの振舞い(出力)を制御するために，包摂するユーザエンティティどうしの相関を包摂構造の知識として用いる．すなわち，ユーザサービスは環境から入力(ユーザからのQoS要求など)が大きく変動した場合，上記包摂構造をメタ知識として包摂するユーザエンティティにメタ入力を作用させる．

2.3.2 ユーザエンティティからアプリケーションエンティティへの包摂構造

ユーザエンティティはユーザサービスを構成するアプリケーションに対応する．一方，アプリケーションレイヤにおいて，アプリケーションは，アプリケーションが利用するリソースに対応したエンティティ(アプリケーションエンティティ)より構成される．したがって，ユーザエンティティは当該アプリケーションのアプリケーションエンティティを包摂する(図6参照)．ユーザエンティティは，包摂するアプリケーションエンティティどうしの相関と包摂するアプリケーションエンティティとの相関を包摂構造の知識として用いる．すなわち，ユーザエンティティはユーザサービスから入力が大きく変動した場合，上記包摂構造をメタ知識として，アプリケーションエンティティにメタ入力を作用させる．このメタ入力はEBAによるサービス間の相互作用の入力と合成されて，当該アプリケーションへのQoS要求を形成する．したがって，ユーザサービスから入力が大きく変動する場合，ユーザエンティティからのアプリケーションQoS要求は大きく変動する．

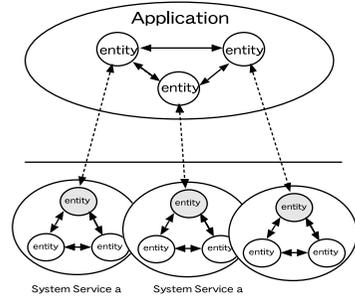


図 11 アプリケーションエンティティからのリソースエンティティとの関連

Fig. 11 The relationship from application entities to resource entities.

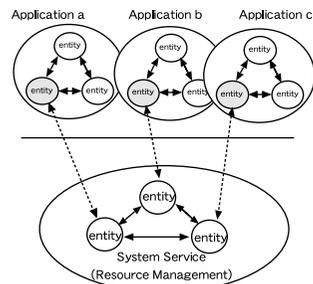


図 12 リソースエンティティからのアプリケーションエンティティとの関連

Fig. 12 The relationship from resource entities to application entities.

2.3.3 アプリケーションエンティティからリソースエンティティへの包摂構造

アプリケーションは，利用するシステムサービスのリソースに対応したアプリケーションエンティティから構成される(図11参照)．一方，システムサービスを構成するエンティティは，当該サービスを利用するアプリケーションに対応する(図12参照)．すなわち，アプリケーションエンティティは，アプリケーションが利用するシステムサービスの当該リソースのエンティティ(リソースエンティティ)と1対1に対応することから，アプリケーションエンティティはリソースエンティティを1対1に包摂する．したがって，アプリケーションエンティティへの入力(アプリケーションQoS要求など)が大きく変動する場合，そのメタ入力により，アプリケーションエンティティからリソースエンティティへの入力(システムQoS要求)が大きく変動する．

2.3.4 システムサービスからのリソースエンティティへの包摂構造

各システムサービス(リソース管理)は，当該リソ

スのリソースエンティティを包摂する．システムサービスは，包摂するリソースエンティティの振舞い(出力)を制御するために，包摂するリソースエンティティどうしの相関とリソース(使用量や使用時間など)とリソースエンティティとの相関を包摂構造の知識として用いる．システムサービスは，外部からの入力(アプリケーションからの QoS 要求やリソースの利用状況など)が大きく変動した場合，包摂するリソースエンティティにメタ入力を作用させる．

ユーザの嗜好性は，ユーザサービスからアプリケーションレイヤまでの包摂構造により入力として作用し，システムサービスにおけるリソース管理制約は，システムサービスからシステムレイヤへの包摂構造により入力として作用する．相互の入力(ユーザの嗜好性とリソース管理制約)は，アプリケーションレイヤとシステムレイヤの 1 対 1 の包摂構造と相互作用メカニズムにより，調整および統合が行われる．

### 3. LAQoS モデルの数理モデル

LAQoS モデルは，EBA に基づいた相互作用のアーキテクチャであること，および EBA 自体も文献 7) においてニューラルネットを用いて数理化されて実験を行っていることから，本章では LAQoS モデルを相互結合ネットワークのニューラルネット<sup>9)</sup>を用いて，数理モデル化を行う．

#### 3.1 エンティティの内部状態と出力

エンティティの内部状態を相互結合ネットワークの素子の活性状態に重ねる．離散時間  $t$  においてレイヤ  $N$  のサービス  $a$  内に含まれエンティティ  $i$  の状態を  $a_i^{S^N}(t)$  と表記する．

また，相互結合ネットワークの素子と同様に，エンティティの内部状態に依存して接続された他のエンティティへ出力を行う．エンティティの外部出力は，エンティティの内部状態のシグモイド関数とし，レイヤ  $N$  のサービス  $a$  に含まれるエンティティ  $i$  の離散時間  $t$  における出力は次のようになる．

$$x_i^{S^N}(t) = \frac{1}{1 + \exp\left(\frac{-a_i^{S^N}(t)}{\epsilon}\right)} \quad (1)$$

したがって，エンティティの出力は 0 から 1 における連続値 ( $0 \leq x_i^{S^N}(t) \leq 1$ ) である．

#### 3.2 エンティティ間の相関

エンティティ間の相関をニューラルネットの相互結合ネットワークにおける素子間の結合重みに重ねる．すなわち，結合重みが正の値である場合はエンティティは互いに協調し，結合重みが負の値である場合はエン

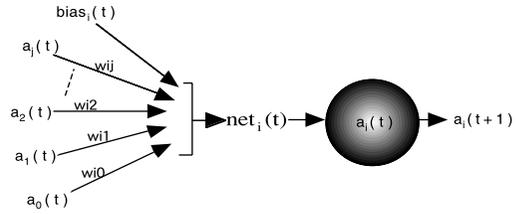


図 13 エンティティへの入力  
Fig. 13 The input to an entity.

ティティは互いに抑制する．レイヤ  $N$  におけるサービス  $a$  に含まれるエンティティ  $i$  と  $j$  の結合重みを  $\omega_{ij}^{S^N}$  と表記する．

#### 3.3 内部メカニズム

エンティティの内部状態の更新方式を内部メカニズムと呼び，エンティティへの入力により更新する規則とする．内部メカニズムは相互結合ネットワークの素子の活性状態の更新則に重ねて，次のようにする．

$$a_i^{S^N}(t+1) = a_i^{S^N}(t) + \begin{cases} (1 - x_i^{S^N}(t))net_i^{S^N}(t) & net_i^{S^N}(t) > 0 \\ x_i^{S^N}(t)net_i^{S^N}(t) & otherwise \end{cases} \quad (2)$$

$net_i^{S^N}(t)$  は，レイヤ  $N$  のサービス  $a$  に含まれるエンティティ  $i$  の離散時間  $t$  における総入力である．相互結合ネットワークの素子と同様に，エンティティへの入力は，接続している他のエンティティの出力と結合重みの積の和により与えられる(図 13 参照)．したがって， $net_i^{S^N}(t)$  は次のようになる．

$$net_i^{S^N}(t) = \sum_{j \neq i} \omega_{ij} x_j^{S^N}(t) + bias_i^{S^N}(t) \quad (3)$$

ただし， $bias_i^{S^N}(t)$  はレイヤ  $N$  のサービス  $a$  に含まれるエンティティ  $i$  の離散時間  $t$  におけるバイアスである．

#### 3.4 エンティティにおけるバイアス

サービスを構成するエンティティにおけるバイアスは，LAQoS モデルにおける包摂構造による上位(サービス利用側)/下位レイヤ(サービス提供側)のエンティティからの入力とする．したがって，レイヤ  $N$  のサービス  $a$  に含まれるエンティティ  $i$  の離散時間  $t$  におけるバイアスは，次のようになる．

$$bias_i^{S^N}(t) = mbias_i^{S^N}(t) + sbias_i^{S^N}(t) \quad (4)$$

ただし，レイヤ  $N$  のサービス  $a$  に含まれるエンティティ  $i$  の離散時間  $t$  において， $mbias_i^{S^N}(t)$  は包摂される上位レイヤのエンティティからのバイアス，

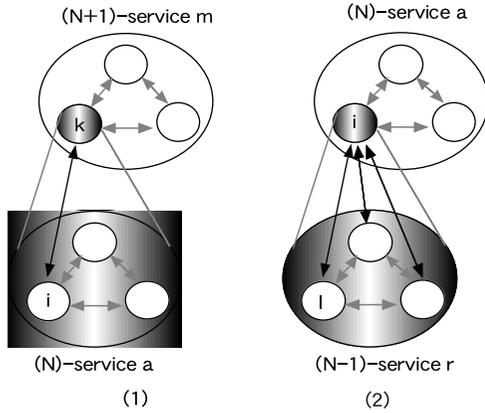


図 14 包摂構造によるレイヤ間のエンティティの結合  
Fig. 14 Connection with entities across layer by subsumption architecture.

$sbias_{i^a}^{S^N}(t)$  は包摂する下位レイヤのエンティティからのバイアスである。

$mbias_{i^a}^{S^N}(t)$  は、包摂構造による上位レイヤのエンティティから当該エンティティへの入力(図 14(1)参照)であることから、次のように定義する。

$$mbias_{i^a}^{S^N}(t) = ebias_{i^a}^{S^N}(t) + pbias_{i^a}^{S^N}(t) \quad (5)$$

ただし、 $ebias_{i^a}^{S^N}(t)$  は当該エンティティ  $i$  を包摂する上位レイヤのエンティティからの EBA に基づく入力、 $pbias_{i^a}^{S^N}(t)$  は当該エンティティ  $i$  を包摂する上位レイヤのエンティティからのメタ入力である。

EBA に基づく入力  $ebias_{i^a}^{S^N}(t)$  は、上位エンティティによる当該エンティティへの相互作用の入力であるので、次のように定義する。

$$ebias_{i^a}^{S^N}(t) = \omega_{ik}^{S^N} x_k^{S^{m+1}}(t) \quad (6)$$

ただし、 $\omega_{ik}^{S^N}$  はレイヤ  $N$  のサービス  $a$  内に含まれるエンティティ  $i$  とレイヤ  $N+1$  のサービス  $m$  内に含まれるエンティティ  $k$  との結合の重みである。

メタ入力  $pbias_{i^a}^{S^N}(t)$  は、当該エンティティを包摂する上位エンティティから、エンティティ間の相互作用メカニズムの動作を変動させる入力である。エンティティ間の相互作用メカニズムは、式 (2) から分かるように、エンティティの総入力に依存する。したがって、LAQoS モデルでは、 $pbias_{i^a}^{S^N}(t)$  を当該エンティティへの総入力の関数(ただし、相互作用による入力とは異なる関数)として仮定し、次のようにする。

$$pbias_{i^a}^{S^N}(t+1) = pbias_{i^a}^{S^N}(t) + z_i^{S^N}(t) net_i^{S^N}(t) \quad (7)$$

ただし、 $z_i^{S^N}(t)$  はレイヤ  $N$  のサービス  $a$  に含まれるエンティティ  $i$  の離散時間  $t$  におけるバイアス重みである。

$sbias_{i^a}^{S^N}(t)$  は、包摂構造による下位レイヤのエンティティから当該エンティティへの入力(図 14(2)参照)であることから、次のように定義する。

$$sbias_{i^a}^{S^N}(t) = \sum \omega_{li}^{S^N} x_l^{S^{r-1}}(t) \quad (8)$$

ただし、 $\omega_{li}^{S^N}$  はレイヤ  $N$  のサービス  $a$  に含まれる当該エンティティ  $i$  と当該エンティティに包摂されるレイヤ  $N-1$  のサービス  $r$  内のエンティティ  $l$  との結合の重みである。

### 3.5 メタ知識

メタ知識は、上位エンティティが、メタ入力を包摂する下位エンティティに作用させるために用いる知識である。この知識には、包摂する下位エンティティにおける相互作用メカニズムの構成情報として、次の知識からなる。

- 包摂するエンティティ間の結合重み  $\omega_{ij}^{S^N}$
- 包摂するエンティティの出力  $x_i^{S^N}(t)$
- 包摂するエンティティとの結合重み  $\omega_{ij}^{S^N}$
- メタ入力  $pbias_{i^a}^{S^N}(t)$  とその重み  $z_i^{S^N}(t)$
- 包摂するエンティティの適応メカニズムにおける評価値  $G^{S^N}(t)$

適応メカニズムの評価値は、ニューラルネット相互結合ネットワークの制約充足度(ネットワークエネルギー関数を負にした関数)に重ねて、次のように求める。

$$G^{S^N}(t) = \sum_{i \neq j} \sum_{j \neq i} \omega_{ij}^{S^N} x_i^{S^N}(t) x_j^{S^N}(t) + \sum_i bias_i^{S^N}(t) x_i^{S^N}(t) \quad (9)$$

この評価値は、マルチメディア処理環境でよく発生するユーザの嗜好とシステムサービスのリソース管理制約が同時に相容れない状況や複数アプリケーション間でリソースを競合する状況などの同時多重制約状態をどの程度充足しているかを判断する客観的数値となる。すなわち、LAQoS モデルでは、価値基準の異なる複数の QoS を全体としてどの程度満たしているかを示す値となる。

## 4. 基本特性

本章では、前章での数理モデルに基づき LAQoS モデルの特性を示す。

#### 4.1 適応メカニズムの基本特性

個々のサービス内におけるエンティティ間の相互作用メカニズムの特性を示すため、 $mbias_i^{S_a^N}(t)$  および  $sbias_i^{S_a^N}(t)$  が無い条件下で、レイヤ  $N$  の任意のサービス  $a$  が離散時間  $t$  においてエンティティ  $i$  の状態を更新した場合について考える。

エンティティ  $i$  がこの更新により、その出力が  $x_i^{S_a^N}(t)$  から  $x_i^{S_a^N}(t+1)$  へ変化した場合、評価値の変化量は式 (9) を用いて次のように求められる。

$$\begin{aligned} \Delta G^{S_a^N}(t+1) &= G^{S_a^N}(t+1) - G^{S_a^N}(t) \\ &= (x_i^{S_a^N}(t+1) - x_i^{S_a^N}(t)) \sum_{j \neq i} \omega_{ij} x_j^{S_a^N}(t) \\ &= (x_i^{S_a^N}(t+1) - x_i^{S_a^N}(t)) net_i^{S_a^N}(t) \end{aligned} \quad (10)$$

式 (10) において、 $x_i^{S_a^N}(t+1) - x_i^{S_a^N}(t) > 0$  である場合、式 (1) により、次の式が成り立つ。

$$a_i^{S_a^N}(t+1) - a_i^{S_a^N}(t) > 0$$

上記の式と式 (2) のエンティティの状態更新則から、次式が得られる。

$$\begin{aligned} a_i^{S_a^N}(t+1) - a_i^{S_a^N}(t) &= net_i^{S_a^N}(t)(1 - x_i^{S_a^N}(t)) > 0 \end{aligned} \quad (11)$$

すなわち、 $net_i^{S_a^N}(t) > 0$  であり、また  $x_i^{S_a^N}(t+1) - x_i^{S_a^N}(t) > 0$  であるので、式 (10) は次式を満たす。

#### 4.2 多階層全体における基本特性

多階層全体において、複数のサービスが混在する場合 (図 15) の多階層全体にわたるメカニズムについて考える。すなわち、 $mbias_i^{S_a^N}$  および  $sbias_i^{S_a^N}$  がある場合を考える。

まず、任意のレイヤ  $N$  の任意のサービス  $a$  が離散時間  $t$  において包摂するエンティティ  $i$  (図 15 参照) の状態を更新した場合の多階層全体の評価値の変化量について考える。エンティティ  $i$  の更新により、離散時間  $t+1$  において、この影響を受けるエンティティは、レイヤ  $N$  のサービス  $a$  内のエンティティ、エン

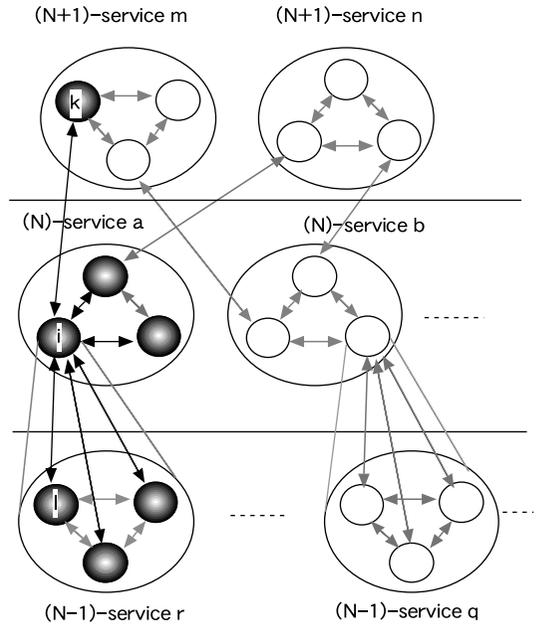


図 15 エンティティの状態更新によるレイヤ間での影響

Fig. 15 Influence across layer when the updating of a state of entity  $i$ .

ティティ  $i$  を包摂する上位レイヤ  $N+1$  のサービス  $m$  のエンティティ  $k$  および下位レイヤ  $N-1$  のサービス  $r$  内のエンティティである。式 (9) を用いてサービス  $a$  の評価値の変化量は次のように求められる。

$$\begin{aligned} \Delta G^{S_a^N}(t+1) &= (x_i^{S_a^N}(t+1) - x_i^{S_a^N}(t)) net_i^{S_a^N}(t) \\ &\quad + z_i^{S_a^N}(t) net_i^{S_a^N}(t) x_i^{S_a^N}(t+1) \end{aligned} \quad (12)$$

ただし、離散時間  $t$  におけるサービス  $a$  のエンティティ  $i$  の総入力  $net_i^{S_a^N}(t)$  は次のようになる。

$$\begin{aligned} net_i^{S_a^N}(t) &= \sum_{j \neq i} \omega_{ij} x_j^{S_a^N}(t) + \omega_{ik}^{S_a^N, S_m^{N+1}} x_k^{S_m^{N+1}}(t) \\ &\quad + \sum_l \omega_{il}^{S_a^N, S_r^{N-1}} x_l^{S_r^{N-1}}(t) + pbias_i^{S_a^N}(t) \end{aligned} \quad (13)$$

また、レイヤ  $N+1$  のサービス  $m$  の評価値の変化量は、エンティティ  $i$  からエンティティ  $k$  への入力のみが変化したので、次のようになる。

$$\begin{aligned} \Delta G^{S_m^{N+1}}(t+1) &= (x_i^{S_a^N}(t+1) - x_i^{S_a^N}(t)) \omega_{ik}^{S_a^N, S_m^{N+1}} x_k^{S_m^{N+1}}(t) \end{aligned} \quad (14)$$

同様に、レイヤ  $N-1$  のサービス  $r$  の評価値の変化量は、エンティティ  $i$  に包摂されるサービス  $r$  内

の各エンティティへの入力に変化したことから、次のようになる。

$$\begin{aligned} & \Delta G^{S_r^{N-1}}(t+1) \\ &= (x_i^{S_a^N}(t+1) - x_i^{S_a^N}(t)) \sum_l \varpi_{il}^{S_a^N S_r^{N-1}} x_l^{S_r^{N-1}}(t) \end{aligned} \quad (15)$$

多階層全体は、レイヤ  $N+1$  のサービス  $m$ 、レイヤ  $N$  のサービス  $a$  およびレイヤ  $N-1$  のサービス  $r$  の包摂構造により、それぞれサービスの適応メカニズムを連結した単層の相互結合ネットワークと見ることができる。したがって、多階層全体の評価値の変化量は次のようになる。

$$\begin{aligned} & \Delta G(t+1) \\ &= \Delta G^{S_m^{N+1}}(t+1) + \Delta G^{S_a^N}(t+1) + \Delta G^{S_r^{N-1}}(t+1) \\ & \quad - (\Delta G^{S_m^{N+1} \cap S_m^N}(t+1) + \Delta G^{S_a^N \cap S_r^{N+1}}(t-1)) \end{aligned} \quad (16)$$

$\Delta G^{S_m^{N+1} \cap S_m^N}(t+1)$  は、エンティティ  $i$  が  $x_i^{S_a^N}(t)$  から  $x_i^{S_a^N}(t+1)$  へ変化した場合において、レイヤ  $N+1$  におけるサービス  $m$  のエンティティ  $k$  とレイヤ  $N$  におけるサービス  $a$  のエンティティ  $i$  との相互結合ネットワークの評価値の変化量であるので、次のようになる。

$$\begin{aligned} & \Delta G^{S_m^{N+1} \cap S_m^N}(t+1) \\ &= (x_i^{S_a^N}(t+1) - x_i^{S_a^N}(t)) \varpi_{ik}^{S_a^N S_m^{N+1}} x_k^{S_m^{N+1}}(t) \end{aligned} \quad (17)$$

同様に、 $\Delta G^{S_a^N \cap S_r^{N+1}}(t-1)$  は、エンティティ  $i$  が  $x_i^{S_a^N}(t)$  から  $x_i^{S_a^N}(t+1)$  へ変化した場合において、レイヤ  $N$  におけるサービス  $a$  のエンティティ  $i$  と包摂するレイヤ  $N-1$  におけるサービス  $r$  の各エンティティとの相互結合ネットワークの評価値の変化量であるので、次のようになる。

$$\begin{aligned} & \Delta G^{S_a^N \cap S_r^{N+1}}(t-1) \\ &= (x_i^{S_a^N}(t+1) - x_i^{S_a^N}(t)) \sum_l \varpi_{il}^{S_a^N S_r^{N-1}} x_l^{S_r^{N-1}}(t) \end{aligned} \quad (18)$$

式(16)に式(12)、(13)、(14)、(15)、(17)、(18)を代入すると、次式が得られる。

$$\Delta G(t+1) = (x_i^{S_a^N}(t+1) - x_i^{S_a^N}(t)) \sum_i \omega_{ij} x_j^{S_a^N}(t)$$

$$+ (pbias_i^{S_a^N}(t+1) - pbias_i^{S_a^N}(t)) x_i^{S_a^N}(t+1)$$

さらに、上記式に、式(7)を適用すると、次式が得られる。

$$\begin{aligned} \Delta G(t+1) &= (x_i^{S_a^N}(t+1) - x_i^{S_a^N}(t)) \sum_i \omega_{ij} x_j^{S_a^N}(t) \\ & \quad + z^{S_a^N} net_i^{S_a^N}(t) x_i^{S_a^N}(t+1) \end{aligned} \quad (19)$$

式(19)において、第1項は前節で示した適応メカニズムの基本特性であるので、評価値を高める方向へ収束する。すなわち、 $\Delta G(t+1)$  は  $z^{S_a^N} net_i^{S_a^N}(t) x_i^{S_a^N}(t+1)$  に依存し、次式が成り立つ。

$$\Delta G(t+1) > z^{S_a^N} net_i^{S_a^N}(t) x_i^{S_a^N}(t+1)$$

ここで、 $\Delta G(t+1)$  の収束条件について考える、評価値が高い方向へ収束する条件は、評価値の変化量が0以上でかつ評価値の変化量の差分が0以下である。したがって、次の2つの式を満たす場合である。

$$z_i^{S_a^N}(t) net_i^{S_a^N}(t) \geq 0 \quad (20)$$

$$\begin{aligned} & z_i^{S_a^N}(t+1) net_i^{S_a^N}(t+1) x_i^{S_a^N}(t+1) \\ & \quad - (z_i^{S_a^N}(t) net_i^{S_a^N}(t)) x_i^{S_a^N}(t) \leq 0 \end{aligned}$$

$$z_i^{S_a^N}(t+1) G^{S_a^N}(t+1) - (z_i^{S_a^N}(t) G^{S_a^N}(t)) \leq 0 \quad (21)$$

まず、上記の条件において、 $x_i^{S_a^N}(t+1) - x_i^{S_a^N}(t) > 0$  である場合について考える、この場合では、前節で示したように、 $net_i^{S_a^N}(t) > 0$  となるので、式(20)は次のようになる。

$$z_i^{S_a^N}(t) \geq 0 \quad (22)$$

また、式(22)が満たされる場合、 $\Delta G^{S_a^N}(t+1) \geq 0$  であるので、式(21)は次のようになる。

$$\begin{aligned} & z_i^{S_a^N}(t+1) \leq \frac{G^{S_a^N}(t)}{G^{S_a^N}(t+1)} z_i^{S_a^N}(t+1) \\ & z_i^{S_a^N}(t+1) \leq z_i^{S_a^N}(t) \end{aligned} \quad (23)$$

また、 $x_i^{S_a^N}(t+1) - x_i^{S_a^N}(t) \leq 0$  においても、同様の議論が成立し、次の式を得る。

$$z_i^{S_a^N}(t) \leq 0 \quad (24)$$

$$z_i^{S_a^N}(t+1) \geq z_i^{S_a^N}(t) \quad (25)$$

式(22)、(23)、(24)および(25)をまとめた次の式

を満たす場合、評価値は高い方向性と収束する。

$$z_i^{S_a^N}(t)net_i^{S_a^N}(t) > 0$$

$$|z_i^{S_a^N}(t+1)| = |z_i^{S_a^N}(t)| - \alpha \quad (26)$$

ただし、 $\alpha > 0$  とする。

以上のことから、階層全体としてが次のような特性を持つ。

- メタ入力が一一定である ( $z_i^{S_a^N}(t) = 0$ ) 場合、またはメタ入力の差分が正の値でその絶対量が減衰する ( $z_i^{S_a^N}(t)net_i^{S_a^N}(t) > 0$  で  $|z_i^{S_a^N}(t+1)| = |z_i^{S_a^N}(t)| - \alpha$ ) 場合、階層全体のメカニズムはその近傍で評価値を高める方向へ収束する (安定状態へ向かう)。
- メタ入力の差分が負の値でその絶対量が増加する ( $z_i^{S_a^N}(t)net_i^{S_a^N}(t) < 0$  で  $|z_i^{S_a^N}(t+1)| = |z_i^{S_a^N}(t)| + \alpha$ ) の場合、評価値は必ずしも上昇せず、階層全体のメカニズムは不安定状態 (新たな状態を探す状態) へ向かう。

したがって、メタ入力において、上記のように  $z_i^{S_a^N}(t)$  を更新することにより、階層全体のメカニズムを安定状態と不安定状態 ( 検索状態 ) 間で遷移させることができる。

### 4.3 メタ入力によるメカニズム制御モデル

#### 4.3.1 不安定状態

前節で証明したように、メタ入力は、階層全体の適応メカニズムの振舞いを変動させることができる。このメタ入力によりメカニズムが不安定となる場合、メカニズムを構成するエンティティの内部状態は多様な状態を動き回る。これをエンティティ  $i$  と  $j$  の間の相互情報量  $M(I, J)$  として表すと、次のようになる。

$$M(I, J) = H(I) + H(J) - H(I, J)$$

$H(I)$  はエンティティ  $i$  の内部状態を事象とした場合のエントロピーで、次のようになる。

$$H(I) = - \sum_t p(a_i^{S_a^N}(t)) \log p(a_i^{S_a^N}(t))$$

ただし、 $p(x)$  は確率密度関数である。したがって、メカニズムを構成するエンティティが多様な状態を動き回る場合、相互情報量は増加する。すなわち、相互情報量が増加する状態は各エンティティのエントロピー ( 乱雑さ ) が増す状態であり、エンティティ間において多様な状態の組合せが生成され、また消滅する状態である。

#### 4.3.2 制御モデル

不安定状態は、エンティティ間において多様な状態の組合せが発生する。この状態への遷移は、当該エン

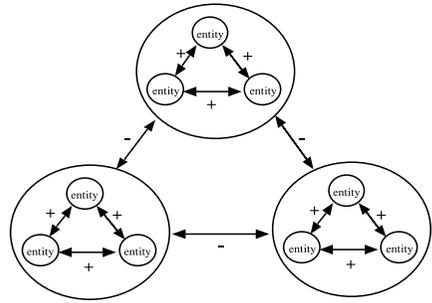


図 16 検証に用いたエンティティの構成  
Fig. 16 Construction of entities for inspection test.

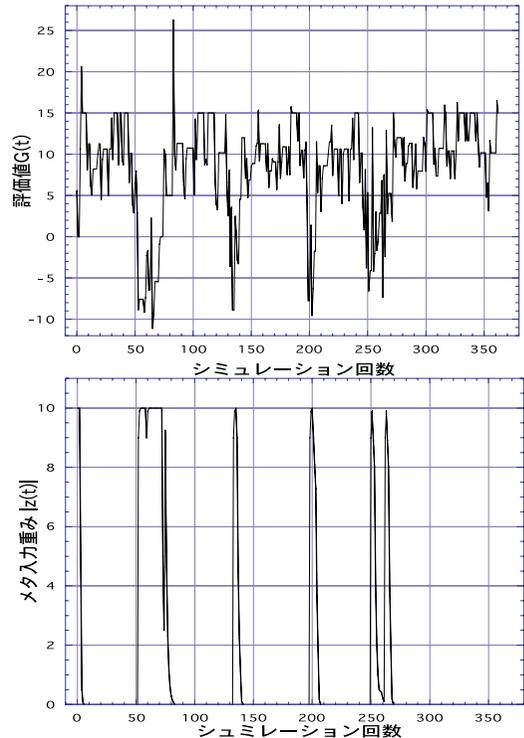


図 17 制御モデルにおけるメタ入力と評価値の関連  
Fig. 17 Relationship with meta input and evaluation value on the control model.

ティティにおいて、その総入力量が大きく変動した場合に行う。総入力量が大きく変動する場合は、環境が大きく変動したことが想定される。この場合に、メタ入力を用いて、メカニズムを不安定な状態に遷移させて、多様な状態の組合せを評価値  $G(t)$  が高くなる条件で検索する。この条件を満たす状態を見つけた場合、再びメタ入力を用いて、メカニズムを安定状態へ遷移させる。

以上のメタ入力による制御では、メタ入力がメタダイナミクス相当の作用を示し、未知の入力に対して新たな振舞いを見つけ出す可能性を高めることとなる。

### 4.3.3 制御モデルの検証

制御モデルの動作を検証するため、メタ入力と評価値との関連に関してシミュレーションを行った。検証データは、3つのエンティティからなるグループを3組用意し、グループ内のエンティティとの相関は協調、グループ外のエンティティとの相関は抑制とした(図16参照)。また、メタ入力は周期的にメタ入力重みを大きくすることにより作用させた。

図17から分かるように、メタ入力重みが大きくなる(メタ入力量が大きくなる)と評価値が短い間に大きく変動する。この状態において、各エンティティの状態の多様な組合せが出現する。この状態から評価値が高くなる状態を見つけるとメタ入力重みを減少させて、評価値を高い値で安定させる動作を行う。このような動作が周期的にメタ入力重みを大きくする作用に合わせて、繰り返し発生する。

以上の結果から、制御モデルは証明された特性に基づき新たな状態を発見していることが判断できる。

## 5. おわりに

本論文では、多階層を構成するQoS適応機能においてそれらを統合する理論的モデルであるLAQoSモデルを提案した。また、提案モデルをニューラルネットを用いて数理化し、さらに、その数理モデルから提案モデルの基本的特性およびネットワークメカニズムの制御モデルを示した。

## 参考文献

- ISO, Quality of Service: Framework, ISO/IEC 13236 (1998).
- Campbell, A., Coulson, G., Gracia, F., Hutchison, D. and Leopold, H.: Integrated Quality of Service for Multimedia Communications, *Proc. IEEE INFOCOM'93*, pp.732-739 (1993).
- Siqueria, F. and Cahill, V.: Quartz: A QoS Architecture for Open Systems, *Proc. IEEE ICDCS'2000* (2000).
- Object Management Group: The Common Request Broker: Architecture and Specification, *OMG Document* (1996).
- Nahrstedt, K. and Smith, J.: Design, Implementation and Experiences of the OMEGA End-Point Architecture, Technical Report (MS-CIS-95-22).
- Brooks, R.A.: Intelligence without representation, *Artificial Intelligence*, Vol.47, pp.139-160 (1991).
- Lambrinos, D. and Scheier, C.: Extended Braintenberg Architectures, AILab Technical Report, No.95-10 (1995).
- Braitenberg, V.: *Vehicles: Experiments in synthetic psychology*, MIT Press (1984).
- Rumelhart, D.E., McClelland, J.L. and the PDP Research Group: *PARALLEL DISTRIBUTED PROCESSING*, The MIT Press (1986).
- Steinmetz, R. and Nahrstedt, K.: *Multimedia: Computing, Communications and Applications*, Prentice Hall PTR (1995).
- 滝沢, 桐本, 倉林, 張, 山崎: 多階層構造による適応的QoSモデルの考察, *信学技法*, CQ2001-36 (2001).
- Ishiguro, A., Shirai, Y., Watanabe, Y. and Uchiyama, Y.: Emergent Construction of Immune networks for autonomous mobile robots through the metadynamics function, *Proc. the Fourth European Conference on Artificial Life*, pp.318-326 (1997).
- Kurabayashi, N., Yamazaki, T., Hasuike, K. and Yuasa, T.: Zinger: Conversation Support Based on Interest Similarity, *International Conference on Multimedia and Expo* (2001).
- 小菅, 山崎, 荻野, 松田: マルチエージェントによる適応的QoS制御方式, *電子情報通信学会論文誌B*, Vol.J82-B, No.5, pp.702-710 (1999).
- Yamazaki, T. and Matsuda, J.: Adaptive QoS Management for Multimedia Application in Heterogenous Environments: A Case Study with Video QoS Mediation, *IEICE Trans. Communications*, Vol.E82-B, No.11, pp.1801-1807 (1999).
- Alexelis, A., Yamazaki, T. and Hasuike, K.: A Distributed Resource Allocation Scheme over DiffServ, toward End-end QoS Assurance, ATR Technical Report, TR-AC-0053 (2001).
- Ogino, N.: A Distributed Multimedia Connection Establishment Scheme in a Competitive Network Environment, *IEICE Trans. Communications*, Vol.E82-B, No.6, pp.819-826 (1999).
- 滝沢, 芝, 大久保: 連続メディア処理における時間制約と通信遅延に適応するタスクスケジューリング, *情報処理論文誌：数理モデル化と応用*, Vol.42, No.SIG5(TOM4), pp.29-41 (2001).
- Zhang, B., Takizawa, Y., Kirimoto, N., Kurabayashi, N. and Yamazaki, T.: Multimedia Communication Coordination based on Layered QoS Model, *ICQR International Workshop on Communication Quality and Reliability*, pp.151-155 (2002).

(平成14年10月31日受付)

(平成14年12月20日再受付)

(平成15年2月8日採録)



滝沢 泰久 (正会員)

1983年京都工芸繊維大学工学部機械工学科卒業。同年日本ユニシス(株)入社。1990年住友金属工業(株)入社。1998年ATR環境適応研究所出向。2002年ATR適応コミュニケーション研究所客員研究員。現在、適応的資源管理方式等の研究に従事。電子情報通信学会, IEEE各会員。



桐本 直樹

1997年関西大学工学部管理工学科(現システムマネジメント工学科)卒業。同年さくらケーシーエス入社。2000年よりATR環境適応通信研究所(現ATR適応コミュニケーション研究所)に出向。現在、ネットワークセキュリティ、適応的QoS制御方式、P2P型アプリケーションに関する研究に従事。電子情報通信学会会員。



倉林 則之 (正会員)

1990年豊橋技術科学大学情報工学課程卒業。1992年同大学院工学研究科情報工学専攻修士課程修了。同年富士ゼロックス入社。1999年から2002年までATR適応コミュニケーション研究所へ出向。現在、富士ゼロックスITメディア研究所研究員。コミュニティ支援およびユーザインタフェースに興味を持つ。



張 兵 (正会員)

1983年中国北京航空航天大学卒業。1990年広島大学大学院博士課程修了。1991年郵政省通信総合研究所(現、独立行政法人通信総合研究所)入所。以来、統計的画像処理、ニューラルネットワークの研究に従事。1995年~1996年米国テネシー大学神経・解剖学科博士研究員。2000年よりATR適応コミュニケーション研究所に出向。現在、適応的QoS制御方式、無線TCPの研究に従事。主任研究員, 工学博士。電子情報通信学会, ACM各会員。



山崎 達也 (正会員)

1987年新潟大学工学部情報工学科卒業。1989年同大学院工学研究科情報工学専攻修士課程修了。同年郵政省通信総合研究所(現、独立行政法人通信総合研究所)入所。1992年~1993年ならびに1995年~1996年カナダ National Optics Institute 客員研究員。1997年より(株)ATR環境適応通信研究所に出向。2001年通信総合研究所けいはんな情報通信融合研究センターに帰任。画像の統計的信号処理, マルチメディア通信におけるサービス品質, ユーザインタフェース等に関する研究に従事。1990年電気関係学会関西支部連合大会奨励賞受賞。博士(工学)。電子情報通信学会, 映像情報メディア学会, 画像電子学会, IEEE各会員。