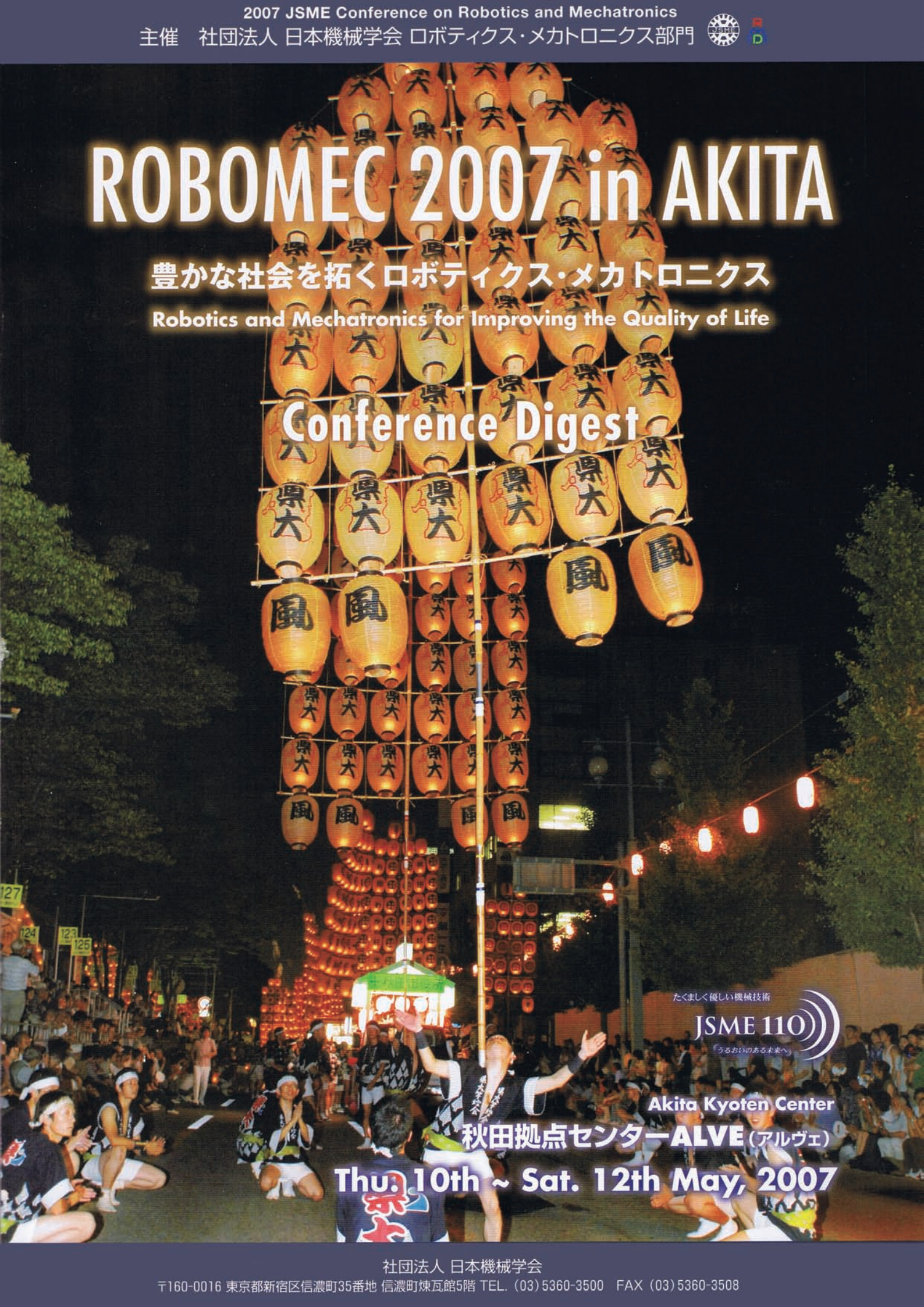


# ROBOMECH 2007 in AKITA

豊かな社会を拓くロボティクス・メカトロニクス

Robotics and Mechatronics for Improving the Quality of Life

## Conference Digest



たくましく優しい機械技術

JSME 110

うるおいのある未来へ

Akita Kyoten Center

秋田拠点センターALVE (アルヴェ)

Thu. 10th ~ Sat. 12th May, 2007

# RT ミドルウェアによる学習・推論コンポーネント

## The learn and reason component on RT-Middleware

○ 玄葉 誠 (アドイン研)      中村 健 (アドイン研)  
正 安藤 慶昭 (産総研)      正 神徳 徹雄 (産総研)

Makoto GENBA, AdIn Research Inc, [genba@adin.co.jp](mailto:genba@adin.co.jp)  
Ken Nakamura, AdIn Research Inc,  
Noriaki ANDO, National Institute of Advanced Industrial Science and Technology  
Tetsuo KOTOKU, National Institute of Advanced Industrial Science and Technology

This paper describes the learning and reasoning software component developed by RT-Middleware called “OpenRTM-aist0.4.0”. It focuses particularly on relative basic technologies, instructions of building component and a system running by using the learning and reasoning component. In addition, this paper is to consider what the software components, such as this learning and reasoning software component, should be in the future of robot system development.

**Key Words:** RT-Middleware, OpenRTM-aist-0.4.0, learn and reason

### 1. はじめに

本発表では、RT ミドルウェアを用いて開発した学習・推論コンポーネントについて、その基盤となる技術、コンポーネント化作業、学習・推論コンポーネントを利用したシステムについて述べる。さらに、今後のロボットシステム開発における、この学習・推論コンポーネントのようなソフトウェア部品の位置付けについて考察する。

### 2. RT ミドルウェア

本開発の基盤となる RT ミドルウェアについて述べる。

#### 2.1 RT ミドルウェア概要

RT ミドルウェアとは、モジュールの組み合わせによりロボットシステムを容易に構築するためのプラットフォームである。RT ミドルウェアの開発は、平成 14 年度から 3 年間、NEDO の 21 世紀ロボットチャレンジプログラム「ロボットの開発基盤となるソフトウェア上の基盤整備」による研究開発プロジェクトとしてスタートした。このプロジェクトの成果として、分散ミドルウェアのインターフェース仕様が策定され、独立行政法人産業技術総合研究所・知能システム研究部門・タスクインテリジェンス研究グループにより、そのプロトタイプ実装である OpenRTM-aist-0.2.0[1]が開発された。

RT ミドルウェアに基づいたロボット開発では、以下のようなメリットがある。

- ・ コンポーネント再利用による工数削減。
- ・ システムの見直し改善、分割/平行開発/デバッグによる開発効率化。
- ・ ネットワーク透過なシステムの実現。
- ・ 言語/OS に非依存な開発環境。

RT ミドルウェアのこうしたメリットを生かすためには、再利用可能な RT コンポーネント群の充実や、多くの利用者/開発者、コンポーネントベンダ、ツールベンダ等の企業の参入が必要であるため、研究開発のみならず、普及活動、標準化活動なども積極的に行っている。

その一環として、RT コンポーネントのインターフェース仕様の標準化をソフトウェア標準化団体 OMG (Object Management Group) において進めており、OMG における 2 年あまりの活動の末、2006 年 9 月 29 日に RT コンポーネント標準仕様 (RTC Specification) が採択され、最終的な文書化の作業に入っている。約 1 年後には、RT コンポーネントの標準仕様が OMG の

公式標準仕様として発行される見込みである。

#### 2.2 OpenRTM-aist-0.4.0

2006 年 9 月 29 日に OMG で採択された、RT コンポーネントの標準仕様に基づいた最新の RT ミドルウェアの実装、OpenRTM-aist-0.4.0 (β版) が、独立行政法人産業技術総合研究所・知能システム研究部門・タスクインテリジェンス研究グループによりリリースされた。

バージョン 0.4.0 では、下記のような点が改良され、標準仕様への準拠およびコンポーネント開発者の利便性が強化されている。

- ・ OMG RTC Specification 準拠。
- ・ データポート (InPort/OutPort) の改善。
- ・ サービスポートの導入。
- ・ 改良された状態遷移の導入。
- ・ 実行主体 (スレッド) の明示的な分離。
- ・ Configuration インターフェースの導入。
- ・ マネージャ機能の強化。

### 3. 学習・推論

本開発の対象とする、学習・推論について述べる。

#### 3.1 学習・推論とは?

“学習”という言葉を知ると、子供が学校で勉強する姿を想像する人も多いかもしれないが、ここでいう“学習”とは人工知能研究の一分野である機械学習のことである。

機械学習とは何か、ウィキペディアの記述 [2] を引用すると“人間が自然に行っている学習能力と同様の機能をコンピュータで実現させるための技術・手法のことである。ある程度の数のサンプルデータ集合を対象に解析を行い、そのデータから有用な規則、ルール、知識表現、判断基準などを抽出する。”となる。この機械学習で得た有用な規則/ルールを“知識”という。

一方、“推論”とは、ある事柄から別の事柄を推し量ることであり、知識をもとに新しい結論を得ることである。

#### 3.2 β-RNA とは

β-RNA は、ユニット間の結合係数としてフィルタ関数 (メンバーシップ関数) を利用したファジィフィルタ型シナプス (Fuzzy Filtered Synapse) によるネットワークモデルを用いた学習・推論エンジンである。

β-RNA の基本機能は、次のものである。

- 複数の入力情報と複数の事象との関係を知識として保持する。(=学習)
- 複数の入力情報が与えられた時、知識によって事象との関係の強さ(合致度)を出力する。(=推論)

### 3.2 $\beta$ -RNA の学習

具体的なイメージが無いとわかりづらいので、カラー画像による果物の識別を例として  $\beta$ -RNA の考え方を説明する。

計測できるデータが、以下の5つであるとする。

- 赤の強度 [red]
- 緑の強度 [green]
- 青の強度 [blue]
- フィレ径(図形を包絡する直方体の辺の平均値) [size]
- 縦横比 [aspect]

識別対象の果物が以下の2つであるとする。

- りんご
- みかん

識別するとは、計測された数値からそれが、“りんご”なのか“みかん”なのかを判断するという事である。この識別を行なう場合、入力情報と出力情報の関係をコンピュータ上でのように表現するかが重要となってくる。もっとも単純に思いつく表現としては、「もし・・・ならば・・・である」という if-then 形式の表現であろう。例えば、「もし色が赤くて少し縦長の直径 8 cm の球形をしているならば、りんごである」とすると、プロダクション・ルールで記述するならば、

if ((色=赤) and (形状=球) and (大きさ=8cm)) then (出力=りんご)

となる。これを計測データと関連づけ、さらにりんごの個体差(値のバラツキ)を考慮すると、次の不等式になる。

if ((red  $\geq$  8.0) and (green  $\leq$  2.0) and (blue  $\leq$  2.0) and (7.0  $\leq$  size  $\leq$  9.0) and (1.0  $\leq$  aspect  $\leq$  1.3)) then (出力=りんご)

ここで懸念されるのは境界領域での問題である。簡単に言えば、size が 6.9 や 9.01 は範囲外とするのか、という点である。そこに“ファジィ”の考え方を適用する。

例えば、“size”を例にとると、不等式による表現は、図 1 の左側のように表現できる。これをファジィ的な表現に置き換えると、図 1 の右側のようなものになる。このようなグラフは値を評価する関数なので評価関数と呼ばれる。ファジィにおいては一般にメンバーシップ関数(Membership Function, MF)と呼ばれる。 $\beta$ -RNA でも、このファジィの表現を用いており、図 1 の右側のようなメンバーシップ関数が内部表現の基本となっている。

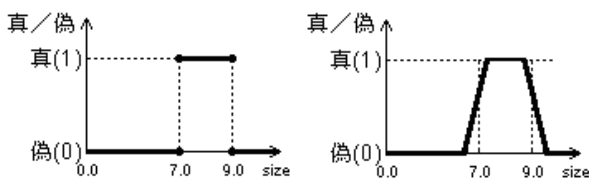


Fig. 1 Inequation and Fuzzy expression

$\beta$ -RNA を用いて、前述の“りんご”を表現する知識を記述すると、図 2 のようになる。このような MF のセットのことを  $\beta$ -RNA ではパターンセット(PS)と呼んでいる。

また、 $\beta$ -RNA では、PS 内の各 MF の相対的な重視度を記述するために、各 MF が“重み”と呼ばれるパラメータを持っている。

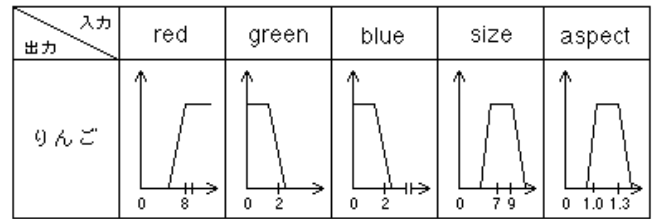


Fig. 2 Pattern Set of Apple

では、“りんご”という集合は  $\beta$ -RNA ではどのように表現されるのであろうか? 前述の PS でも“りんご”を表現することはできるが、“りんご”の中にも“ふじ”, “紅玉”などの種類がありこれらの集合が“りんご”となる。そういう構造を記述するために、通常  $\beta$ -RNA では、“りんご”など識別対象をパターンテーブル(PT)と呼ばれる PS の集合で記述する。そして、“りんご”, “みかん”という複数の PT と、入力情報(計測データ)を記述する枠組みであるセンサセット(SS)によってパターングループ(PG)が構成される。

以上が  $\beta$ -RNA での知識表現の基本的な形態である。

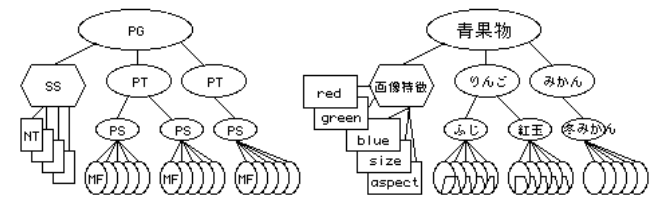


Fig. 3 Data Structure of  $\beta$ -RNA

このような知識表現で、入力データや識別対象などが非常に多くなり記述すべき知識(MF)が膨大になると記述もメンテナンスも辛いものになる。そこで  $\beta$ -RNA では、入力データに対応した出力(識別対象)の正解を与えることで自動的に MF が生成される方式をとる。具体的には、入力された教師データから度数分布を作成しそれを正規分布へ近似したものを MF として作成する。さらに入力データに応じて PS を自動的に拡張していく機能も有する。これが  $\beta$ -RNA における学習である。

### 3.2 $\beta$ -RNA の推論

$\beta$ -RNA における推論とは合致度を求めることである。合致度とは入力データと知識として記述されている事象との間の合致(適合, 符合)の度合を示す指標である。例えば計測されたデータ(色, 形状など)が“りんご”という事象にどれだけ当てはまるか(何%合致するか)ということ量を指標として求めることが  $\beta$ -RNA の推論である。

前項で、MF, PS, PT, SS という構造は示した。出力として合致度を持つのは、MF, PS, PT である。各々の合致度算出の方法を述べる。

まず SS によって正規化された入力データが示す座標(横軸)の MF の高さ(縦軸上の値)が、MF の合致度として得られる。

PS における合致度は、PS 内の MF の合致度を合成することで得られる。合成には、単純相和平均、単純相乗平均、重み付き相和平均、重み付き相乗平均などのいくつかの方式が定義されており、その中から指定された方式で算出される。

PT の合致度は、その PT 内の全 PS の合致度の中で最大のものとなる。論理的には、PT 内の PS は OR 結合されているということになる。

### 3.2 $\beta$ -RNA のネットワーク表現

前項で述べた  $\beta$ -RNA の推論方式を、処理のデータフローを基にネットワークモデル化すると図 4 のようなモデルになる。

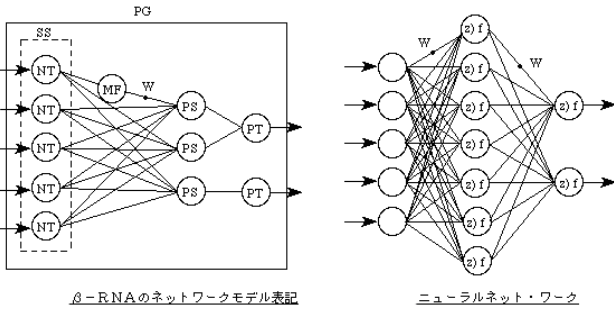


Fig. 4 Network model of  $\beta$ -RNA and Neural Network

3層(中間層1層)のニューラルネットワークモデルと比較的似た構造をしているが、ニューラルネットワークが2層の等価なネットワークによって構成されているのに対して、 $\beta$ -RNAは前段の層で殆どの機能を満たし、後段のネットワークが補助的なものになっている点が大きく異なる。

この $\beta$ -RNAのネットワークモデルは、MFというフィルタ関数と重みを持った素子(リンク+ノード)を基本構成要素としている。この基本構成要素のことをFFS(Fuzzy Filtered Synapse:ファジィフィルター型シナプス)と言う。

$\beta$ -RNAに関する詳細な説明は、文献[3][4]を参照されたい。

#### 4. 学習・推論コンポーネントの実現

RTミドルウェアを用いて学習・推論エンジン $\beta$ -RNAをRTコンポーネント化する作業について述べる。

##### 4.1 設計

まず始めに $\beta$ -RNAの動作について整理する。 $\beta$ -RNAはC言語で記述されたライブラリであり、関数インターフェースが提供される。 $\beta$ -RNAの動作は学習と推論という2つの動作がある。それぞれの動作に対応した関数提供され、それを順番にコールする。

学習モードの動作は、下記ようになる。

- ・ 初期化処理
- ↓
- ・ パラメータファイルの読み込み
- ↓
- ・ 入力数値と教示信号による学習
- ↓
- ・ 知識データ(学習結果)の保存
- ↓
- ・ 終了処理

推論モードの動作もほぼ同様のシーケンスであり、下記ようになる。

- ・ 初期化処理
- ↓
- ・ パラメータファイルの読み込み
- ↓
- ・ 知識データ(学習結果)の読み込み
- ↓
- ・ 入力数値による推論
- ↓
- ・ 終了処理

RTコンポーネントもC++で記述できるので、RTコンポーネントの状態遷移やPortのデータ操作に対応して各関数の呼出を行えば、特に大きな問題はないと考えられる。なお、

RTコンポーネントの状態遷移は、OpenRTM-aist-0.2.0からOpenRTM-aist-0.4.0へのバージョンアップで変更され、よりシンプルになっている。表1に、RTコンポーネントの状態遷移に合わせて行う $\beta$ -RNAの動作を示す。

Table 1  $\beta$ -RNA Action of Component method

	Component method	$\beta$ -RNAの処理
学習	onStartup()	初期化処理
	onExecute()	パラメータファイル読み込み 学習実行 知識データ保存
	onShutdown()	終了処理
推論	onStartup()	初期化処理
	onExecute()	パラメータファイル読み込み 知識データ読み込み 推論実行
	onShutdown()	終了処理

次に、RTコンポーネントのPortの定義を検討する。

RTコンポーネントには、InPort/OutPort/ServicePortの3種類のポートが用意されている。InPort/OutPortは、OpenRTM-aist-0.2.0から存在するPortで、データの受渡しを行う事を目的に設計されている。ServicePortはOpenRTM-aist-0.4.0で追加されたPortで、メソッド呼出しを行う事ができる。

学習・推論コンポーネントの動作は、ロボットアームの制御などと異なり連続的に動作するものではない。利用者(他コンポーネント)の要求に応じて入力に対する結果を算出し出力するという動作である。このような動作の実現は、InPort/OutPortでも、ServicePortでも、どちらの方式でも可能である。

InPort/OutPortを使った場合は、RTコンポーネントのActive状態で周期的にコールされるメソッドonExecute()の中でPortのデータが更新されたか否かをチェックし、更新された場合のみ対応する関数を呼出せばよい。

InPort/OutPortを使った学習・推論コンポーネントのポート構成の例を表1に示す。

Table 2 InPort/OutPort of learn and reason component

	Name	DataType	Note
In	DataCount	TimedShort	入力指標数
	Datas	TimedDoubleSeq	入力データ
	ID	TimedShort	識別ID
	ParamDir	TimedString	パラメータディレクトリ
Out	ID	TimedShort	識別ID
	AccordRatio	TimedDouble	合致度

一方、ServicePortを使った場合、サービスを提供するプロバイダ側は自身のメソッドとして提供する処理を実装する。サービスコールするクライアント側は、RTミドルウェアのフレームワークによりサービスのオブジェクト参照が設定されているので、それを自身のメソッドを呼ぶのと同様に必要に応じてコールすれば処理実行される。

ServicePort を使った実装では、表 1 に示したデータがメソッドの引数となる。

### 4.3 実装

RT コンポーネントの状態遷移と Port の定義が決まってしまうと、その実装は比較的容易である。

まず、RT ミドルウェアが提供するコマンド `rtc-template` を実行し、RT コンポーネントのテンプレートを作成する。このコマンドの実行で、必要なソースファイル、ヘッダファイル、Makefile などが作成される。

あとは状態遷移に対応したメソッドのコメントアウトを外して、必要な処理（学習・推論コンポーネントの場合は  $\beta$ -RNA が提供する関数のコール）を記述していけばよい。

各メソッドの実装が終了すれば、`make` の実行だけで、単体実行できる形式、および共有ライブラリ形式の RT コンポーネントが作成される。

## 5. 学習・推論コンポーネントの利用

学習・推論コンポーネントの利用例について述べる。

### 5.1 話者認識システム

学習・推論コンポーネントを利用したシステムの一例として、人間の音声から、誰の声を判定する話者認識システムを実現した。話者認識システムは下記の 6 つのコンポーネントで構成される。

☆ 音声入力コンポーネント

音声をマイクやファイルから入力する。

☆ 特徴量抽出コンポーネント

音声データを FFT 解析し、パワースペクトルを求める。

☆ 学習・推論コンポーネント

$\beta$ -RNA による学習・推論コンポーネント

☆ 学習・推論コントロールコンポーネント

話者名やパラメータファイル名を入力する。

☆ 話者管理コンポーネント

識別 ID と話者名の関係付けを管理する。

☆ 結果表示コンポーネント

推論結果を表示する。

これらの各コンポーネント起動し、RT ミドルウェアが提供する `RtcLink` で Port を連結して動作させた状態を下図に示す。

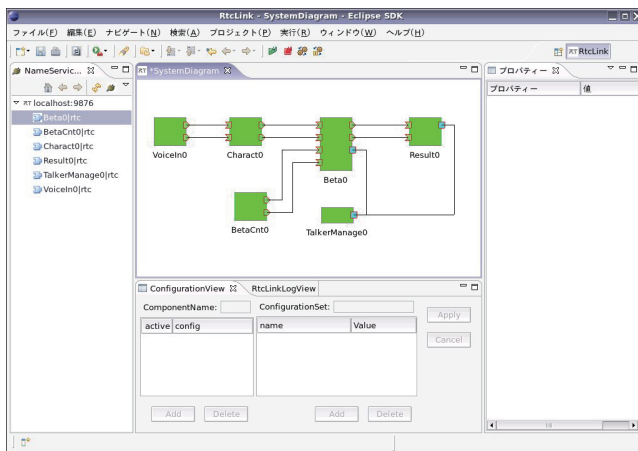


Fig. 5 Speaker recognition system on RT-Middleware

このシステムの動作状況、話者認識の精度などは、ぜひ Robomec' 07 の会場で確かめていただきたい。

## 5.2 その他のシステム

本発表ではパターン認識の一例として話者認識システムを実現したが、 $\beta$ -RNA は「数値データを入力とする識別器」であるので、音声に限らず数値で表現されるものであればそれらの関係を学習し推論することができる。今までに  $\beta$ -RNA を利用したコンピュータシステムとしては下記のものがある。

- ・ 建設ロボットの制御システム
- ・ トンネル掘削機方向・姿勢制御システム
- ・ 工作機械・適応制御システム
- ・ 工場生産ライン向け検査システム
- ・ 装置系の故障及び故障予兆の検知

このようなシステムを、RT ミドルウェアを用いて構築する際には学習・推論コンポーネントが有効に使えるであろう。

また、経済産業省が推進する次世代ロボットプロジェクトでは「ロボットが環境・状況認識能力を有し、自律的な判断能力を持つ知能化技術を開発することが必要」という技術ニーズが掲げられている。学習・推論コンポーネントは、環境状況認識や自律判断機能の実現へ向けての第一歩だと考えている。今後、より一般的な機能構成、あるいはより高度な判断機能の実現にむけて研究開発を進めていきたい。

## 6. ソフトウェアコンポーネントに対する考察

学習・推論コンポーネントのような純粋にソフトウェアのみのコンポーネントで重要なのは、コンポーネントの独立性、汎用性をどこまで実現できるかという点である。

ソフトウェアの世界で“部品化”、“再利用”というとなかなか難しいが、RT コンポーネントは RT ミドルウェアというフレームワーク上に構築するので、“部品化”自体は難しくはない。

しかし、ハードウェアに絡んだ制約がないので自由度が高い反面、ある一連の機能をどこで複数のコンポーネントに分割するか、どのような Port 定義にするか、といった点が重要となってくる。

例えば、RT コンポーネントの `InPort/OutPort` を使ってデータ授受のみで他コンポーネントと連携するの状態であればコンポーネントの独立性、汎用性も高くなる。一方、`ServicePort` を使うと任意の時点でメソッド呼出しができソフトウェア的に実現できる事も多くなるが、コンポーネントの独立性としては低くなる。

これらの点を目的に合わせて上手く設計する必要がある。

## 7. おわりに

本発表では、RT ミドルウェアを用いて開発した学習・推論コンポーネントについて、その基盤となる技術、コンポーネント化作業、学習・推論コンポーネントを利用したシステムについて述べた。

今後、RT ミドルウェアが普及し、相互に接続できる RT コンポーネントが増えてくればロボットシステムの構築しやすさは飛躍的に向上すると期待される。その為にも RT ミドルウェアの標準化や改善、様々な RT コンポーネントの開発を続けていきたい。

## 文献

- [1] OpenRTM-aist チュートリアル  
URL: <http://www.is.aist.go.jp/rt/OpenRTM-aist-Tutorial/index.html>
- [2] 「機械学習」『フリー百科事典 ウィキペディア日本語版』2007年1月26日 14:45 (UTC)版 URL: <http://ja.wikipedia.org>
- [3] 中村, 上石, "多入力/多出力制御でリアルタイムの推論処理と知識獲得を実現したファジィ推論ツール", 日経 AI 別冊 1990 年春号, pp.152-163, 日経 BP, 1990.
- [4] 中村, 藤巻, 渡辺, 浜谷, 上石, "FFS-Network を用いた連続値型学習/推論システム", 第9回ファジィシステムシンポジウム, 1993.