

R T Mによるコンポーネント化作業

学習・推論コンポーネントを用いた
話者識別システム

運用マニュアル

平成 19 年 05 月 31 日

株式会社アドイン研究所

【目次】

1. はじめに	3
2. 話者識別システム概要	4
2.1. 機能概要	4
2.2. 学習・推論コンポーネント	4
3. 稼働環境	6
4. インストール、システム設定	7
4.1. インストール方法	7
4.2. システムの設定方法	7
5. RT コンポーネント一覧	9
6. 話者識別システムの操作方法	11
6.1. CUI 版話者識別システム	11
6.1.1. 起動	11
6.1.2. システム構築	11
6.1.3. 操作	13
6.1.4. 終了	19
6.2. GUI 版話者識別システム	20
6.2.1. 起動	20
6.2.2. システム構築	20
6.2.3. 操作	23
6.2.4. 終了	25
7. コンポーネントのインターフェース仕様	26
7.1. インターフェース仕様	26
7.1.1. 音声入力コンポーネント（マイク版）	26
7.1.2. 音声入力コンポーネント（ファイル版）	26
7.1.3. 話者管理コンポーネント	27
7.1.4. 音声特徴量抽出コンポーネント	27
7.1.5. 学習・推論コンポーネント（データポート版）	28
7.1.6. 学習・推論コンポーネント（サービスポート版）	28
7.1.7. ユーザインターフェースコンポーネント（CUI 版）	28
7.1.8. ユーザインターフェースコンポーネント（GUI 版）	29
7.2. コンポーネント間シーケンス	31
7.2.1. CUI 版話者識別システム	31
7.2.2. GUI 版話者識別システム	32
8. コンポーネント設定ファイル仕様	33
8.1. 音声特徴量抽出コンポーネントの設定ファイル	33
8.2. 音声入力コンポーネント（マイク版）の設定ファイル	34

8.3. 学習・推論コンポーネントの設定ファイル	35
9. その他.....	36
9.1. ライセンス等.....	36
9.2. 連絡先.....	36

1. はじめに

本書は、RT コンポーネント「学習・推論コンポーネント」を用いた話者識別システムの運用マニュアルである。

「学習・推論コンポーネント」の詳細は、別紙『「学習推論コンポーネント」ユーザマニュアル』を参照のこと。

2. 話者識別システム概要

2.1. 機能概要

話者識別システムは、入力音声が入力され、予め登録してある話者のうち、「誰により、発声されたものか？」を判定するシステムである。以下の2つの機能が存在する。

機能1) 音声を登録する (= 学習機能)

話者識別システムに、音声と、その発話者とを、複数回入力する。話者識別システムは、入力された音声の特徴を解析して、発話者の音声の特徴を、学習する。

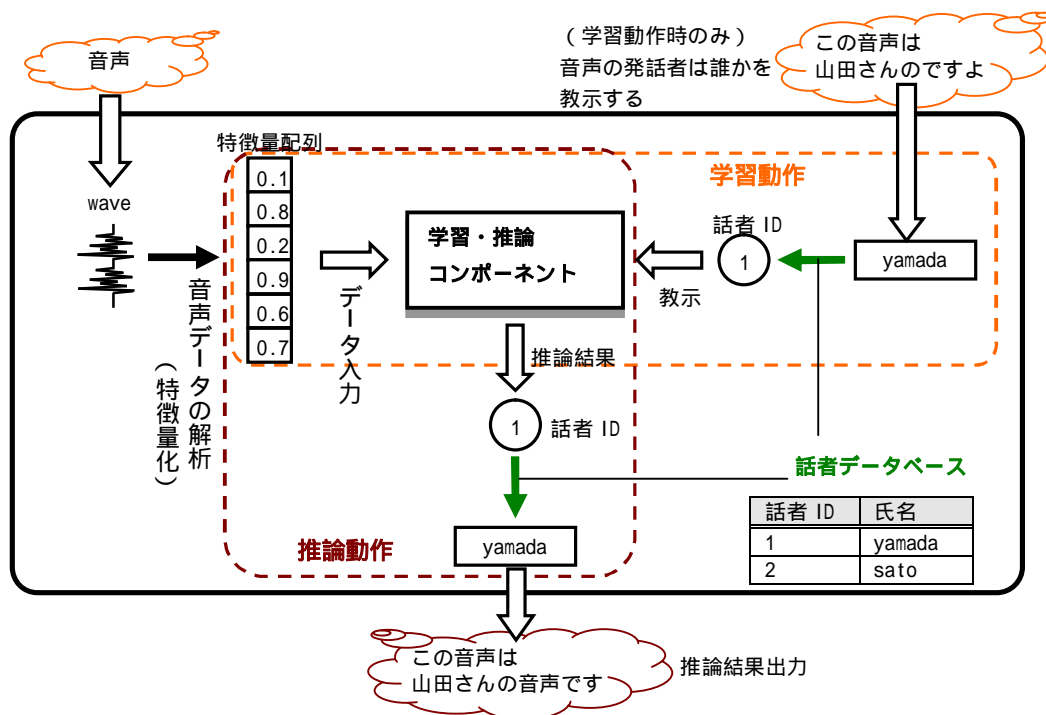
機能2) 誰により発音されたものかを判定する (= 推論機能)

話者識別システムに音声を入力すると、話者識別システムは、学習した結果を元に、その音声は誰のものかを推論して、その結果を返す。

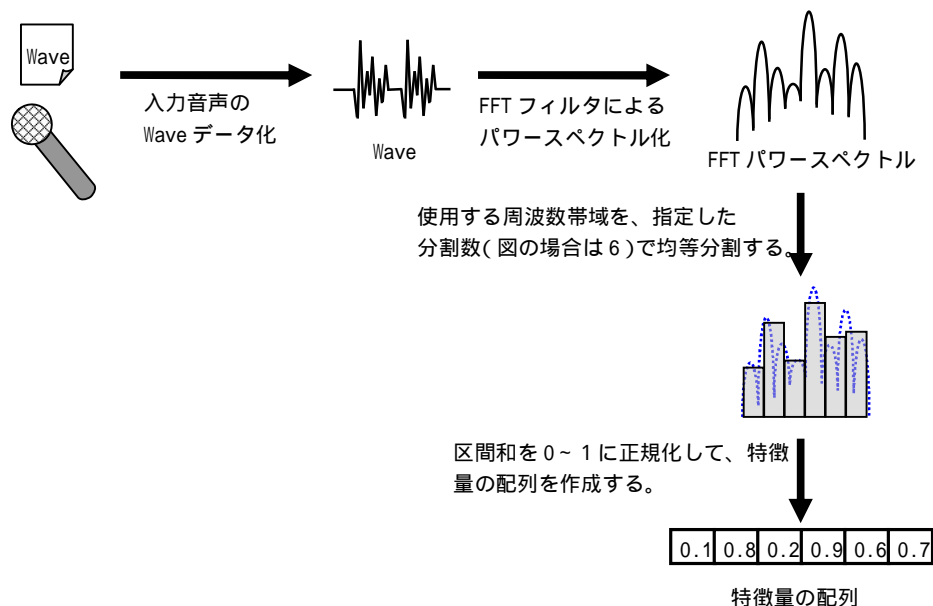
なお、本システムへの音声入力は、(1)wave ファイルによる音声入力、(2)マイクによる音声入力、のいずれかにより行う。

2.2. 学習・推論コンポーネント

話者識別システムは、汎用的な学習・推論機能を提供する RT コンポーネント「学習・推論コンポーネント」を利用している。話者識別システム内で、「学習・推論コンポーネント」が果たす役割を、下図に示す。



学習・推論コンポーネントは、特徴量 (= 0~1 の数値) の配列を入力とする (= 音声データを、直接、入力することはできない)。そのため、話者識別システムでは、入力された音声データを特徴量化する。特徴量化のアルゴリズムを、下図に示す。



学習・推論コンポーネントは、話者識別システムの実行される機能に応じて、学習動作と、推論動作を行う。各々の動作内容を、以下に示す。

学習動作： 入力された特徴量データと、話者 ID との関係を、学習する。

推論動作： で学習した結果を元に、新たに入力された特徴量データが、どの話者 ID のものかを推論して、結果を出力する。

なお、学習・推論コンポーネントは、話者名 (= 文字列データ) を直接扱うことができない。そのため、話者識別システムは、システム内部において、話者に ID (0 以上の整数) を割り当て、話者を ID で管理している。

3. 稼働環境

本システムは、RT ミドルウェア上で動作する。開発環境、動作対象環境、動作確認済み環境を、以下に示す。

開発環境

- ・ Vine-Linux 3.2 (gcc バージョン 3.3.2 でコンパイル)
- ・ RT ミドルウェア「OpenRTM-aist-0.4.0-RC2」

動作対象環境

- ・ gcc バージョン 3.3.2 以上がインストールされている Linux(x86)環境
- ・ RT ミドルウェア「OpenRTM-aist-0.4.0-RC2」

動作確認済み環境

- ・ Vine-Linux 3.2
- ・ RT ミドルウェア「OpenRTM-aist-0.4.0-RC2」

本システムの実行に必要なソフトウェアを、以下に示す。

- | |
|--|
| <ul style="list-style-type: none">・ RT ミドルウェア
(ランタイムライブラリ、RTCLink、CORBA 製品 (omniORB 等) を含む)・ Julius adintool (音声入力ツール) (*1)・ GTK+2.0 ランタイムライブラリ (*2) |
|--|

(*1) 声認識システムの開発・研究のための、高性能な汎用大語彙連続音声認識エンジン。オープンソースである。話者識別システムでは、マイクからの音声入力を可能にするために、使用している (音声認識機能は、使用していない)。詳細は、julius の web ページ(<http://julius.sourceforge.jp/>)を参照のこと。

(*2) GUI 版の話者識別システムを使用する場合のみ、必要となる。CUI 版の話者識別システムを使用する場合は、不要。

4. インストール、システム設定

4.1. インストール方法

(1) RT ミドルウェア「OpenRTM-aist-0.4.0」のインストール

下記のページに従い、RT ミドルウェア「OpenRTM-aist-0.4.0」のインストールを行う。なお、既に、「OpenRTM-aist-0.4.0」がインストールされている場合は、再度、インストールを行う必要はない。

<http://www.is.aist.go.jp/rt/OpenRTM-aist/html/>

(2) 音声入力ツール「Julius adintool」のインストール

下記のページに従い、音声入力ツール「Julius adintool」のインストールを行う。なお、既に、「Julius adintool」がインストールされている場合は、再度、インストールを行う必要はない。

<http://julius.sourceforge.jp/>

(3) GTK+2.0 のインストール

Gtk+2.0 以上がインストールされていない場合は、下記ページに従い、Gtk+2.0 のインストールを行う。

<http://www.gtk.org/>

(4) 話者識別システムのインストール

CD-ROM の **src** 以下の**ファイル一式**を、インストール先のディレクトリにコピーする。次に、スクリプトファイル "インストールディレクトリ/**build.sh**" を実行する。

4.2. システムの設定方法

話者識別システムを起動する前に、システム設定を行う必要がある。以下の手順に従い、設定ファイルの必要な項目に、値を設定する。

(1) 音声認識ツール「Julius adintool」のパスを設定する

設定ファイル "インストールディレクトリ/run/jr.ini" の項目 **AdintoolPath** に、4.1 章の(2)でインストールした音声入力ツール「Julius adintool」のインストールパスを設定する。

例) 設定ファイル "インストールディレクトリ/run/jr.ini" の項目 **AdintoolPath**

AdintoolPath=/usr/local/bin/adintool

(2) ポート番号を設定する

設定ファイル "インストールディレクトリ/run/jr.ini" の項目 **PortNo** に、音声入力ツール「Julius adintool」と、話者識別システムが通信を行うためのポート番号を設定する（49152 以上 65535 以下のポート番号の中で、他のアプリケーションが使用していない番号を指定すればよい）。

例) 設定ファイル "インストールディレクトリ/run/jr.ini" の項目 **PortNo**

```
PortNo=49159
```

(3) サウンドデバイスのサンプリングレートを設定する

設定ファイル "インストールディレクトリ/run/jr.ini" の項目 **SamplingRate** 及び、設定ファイル "インストールディレクトリ/run/ps.ini" の項目 **SamplingFreq** に、サウンドデバイスのサンプリングレートを設定する。

例) 設定ファイル "インストールディレクトリ/run/jr.ini" の項目 **SamplingRate**

```
SamplingRate=16000.000000
```

例) 設定ファイル "インストールディレクトリ/run/ps.ini" の項目 **SamplingFreq**

```
SamplingFreq=16000.000000
```

(4) RT コンポーネントのネームサーバを指定する

RT コンポーネント設定ファイル "インストールディレクトリ/run/rtc.conf" の項目 **corba.nameservers** に、話者識別システムを構成する RT コンポーネントを登録するためのネームサーバ（アドレス：ポート番号）を指定する。

例) 設定ファイル "インストールディレクトリ/run/rtc.conf" の項目

corba.nameservers

```
corba.nameservers:localhost.localdomain:9876
```

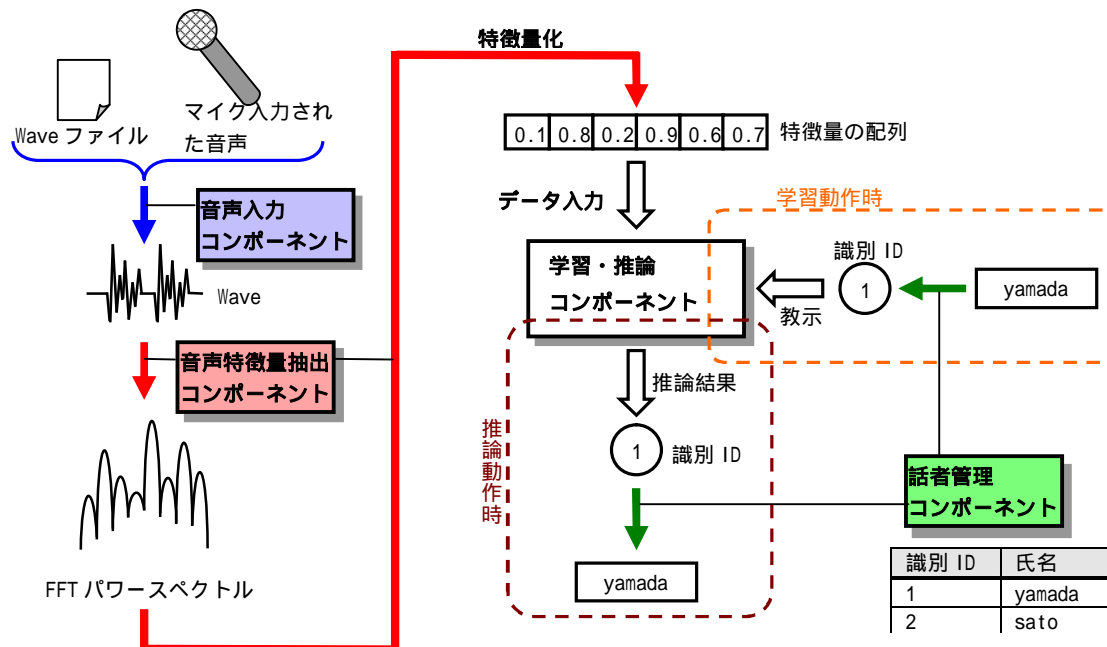
話者識別システムのシステム設定は、以上である（設定ファイル内の、他の項目は、デフォルト値のままでよい）。なお、設定ファイルの各項目の意味は、『8章 コンポーネント設定ファイル』を参照のこと。

5. RT コンポーネント一覧

話者識別システムで使用する、RT コンポーネントの一覧を以下に示す。

コンポーネントの種類	コンポーネント名	機能
音声入力コンポーネント (マイク版)	VoiceInMic	入力開始トリガを受けて、マイク入力された音声を Wave 形式で出力する。(内部で、音声入力ツール『Julius addtool』を使用している。)
音声入力コンポーネント (ファイル版)	VoiceInFile	入力されたファイルパスに格納されているデータを、Wave 形式で出力する。
話者管理 コンポーネント	TalkerManage	話者名と話者 ID (話者に割り振られる ID) の関係を管理する。
音声特徴量抽出 コンポーネント	Charact	入力された Wave データを FFT 解析し、パワースペクトルを求め、周波数帯別に平均した値(これを音声の特徴量とする)を出力する。
学習・推論 コンポーネント	BetaRnaService	入力された音声の特徴量と、話者 ID との関係を学習する。 入力された音声の特徴量が、どの話者 ID に属するものか (= 話者が誰か) を推論する。

上記の各 RT コンポーネントが、話者識別システムの中で果たす役割を、図示する。



話者識別システムでは、上記の RT コンポーネントのほか、ユーザの入力を受け付ける/ユーザに情報を出力する役割の RT コンポーネントが存在する。

コンポーネントの種類	コンポーネント名	機能
インターフェース コンポーネント(CUI 版)	CUI	ユーザに、話者識別システムのコンソールインターフェース(CUI)を提供する。
インターフェース コンポーネント(GUI 版)	GUI	ユーザに、話者識別システムのグラフィカルなインターフェースを提供する。

6. 話者識別システムの操作方法

話者識別システムは、コンソール（CUI）により操作を行うシステム（以下、CUI 版話者識別システムと呼ぶ）と、GUI により操作を行うシステム（以下、GUI 版話者識別システムと呼ぶ）との2通りを用意している。CUI 版話者識別システムは、データポートを使用する学習・推論コンポーネントのサンプルシステムとなっており、GUI 版話者識別システムは、サービスポートを使用する学習・推論コンポーネントのサンプルシステムとなっている。

なお、GUI 版話者識別システムは、GTK+2.0 以上が使用可能な環境でのみ、動作する。従って、この環境を満たさない場合は、CUI 版話者識別システムを実行することになる。

6.1. CUI 版話者識別システム

6.1.1. 起動

スクリプトファイル "インストールディレクトリ/run/cui.sh" を実行する。これにより、話者識別システムを構成する RT コンポーネントが、全て起動する。画面には、6 つのコンソールウィンドウが表示される。この状態では、各 RT コンポーネントは、全てアイドル状態である。

6.1.2. システム構築

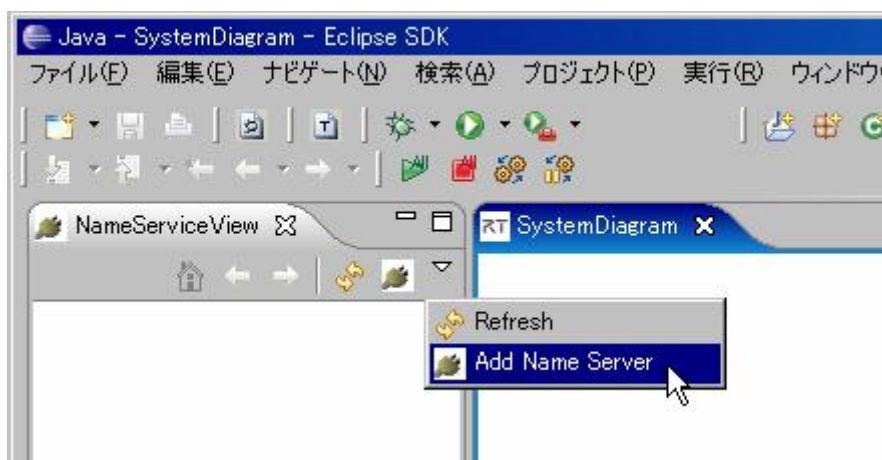
(1) RTCLink 起動する。

RTCLink の起動、操作方法は、以下のページを参照のこと。

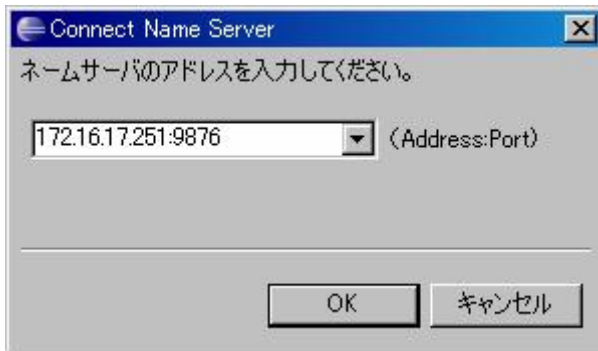
<http://www.is.aist.go.jp/rt/OpenRTM-aist/html/FrontPage.html>

(2) ネームサーバに接続する。

NameServiceView から、[AddNameServer]メニューを選択する（下図参照）。

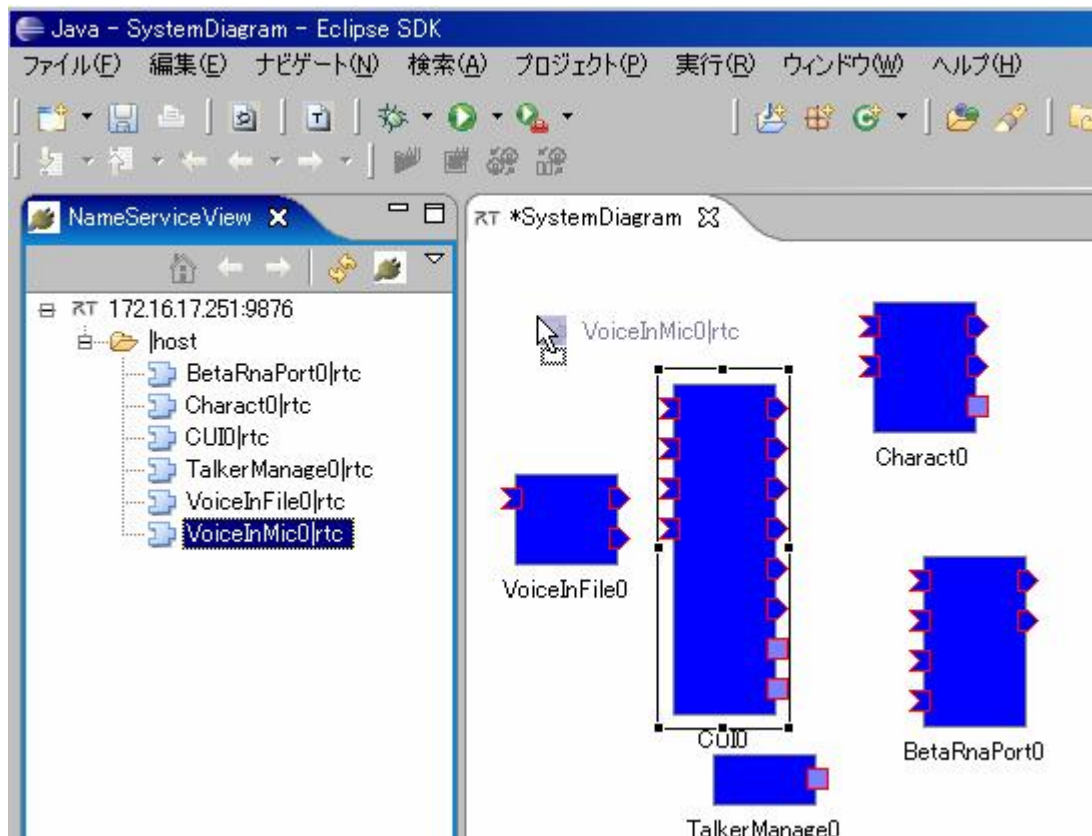


ConnectNameServer 画面で、ネームサーバを指定する（下図参照）。ネームサーバのアドレス、ポート番号は、設定ファイル "インストールディレクトリ/run/rtc.conf" の項目 **corba.nameservers** で指定したものを記述すればよい。

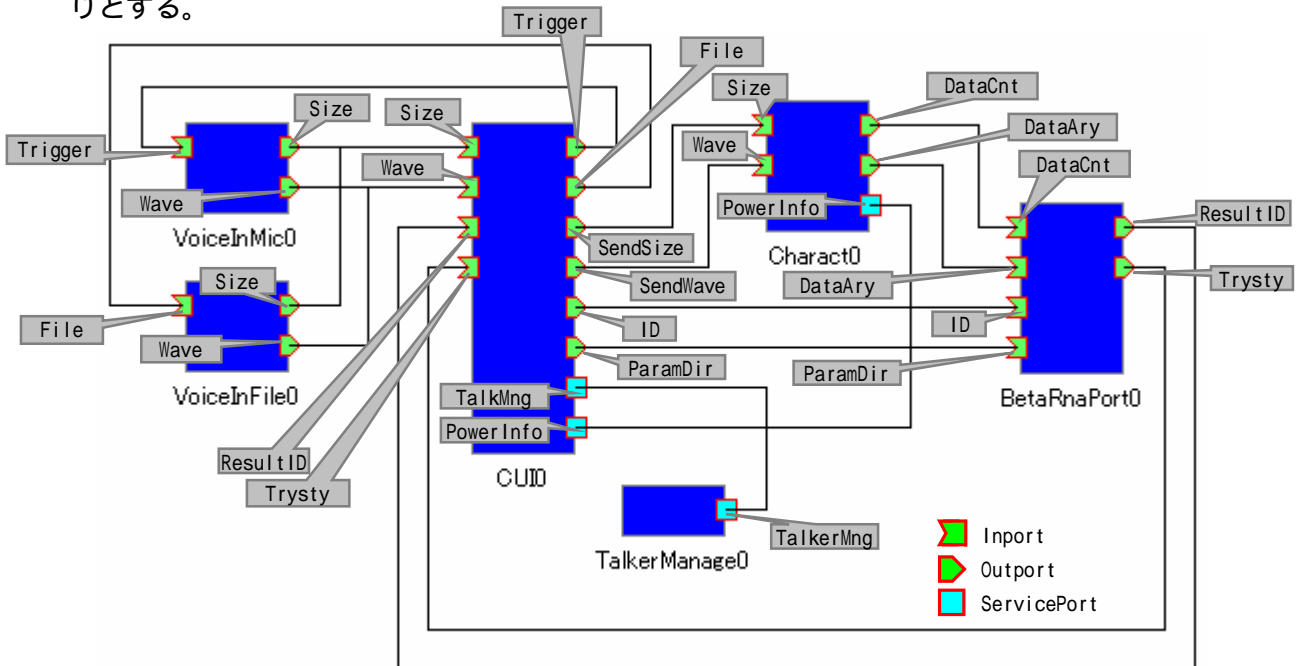


(3) RT コンポーネントの接続を行う

[ファイル]-[OpenNewSystemEditor]メニューを選択する。SystemDiagram ウィンドウが作成されるので、そこに、NameServiceView で表示されている RT コンポーネントを、全て、ドラッグアンドドロップ操作により、配置する。



次に、配置した RT コンポーネントのポート同士を接続する。ポートの接続は、下図の通りとする。



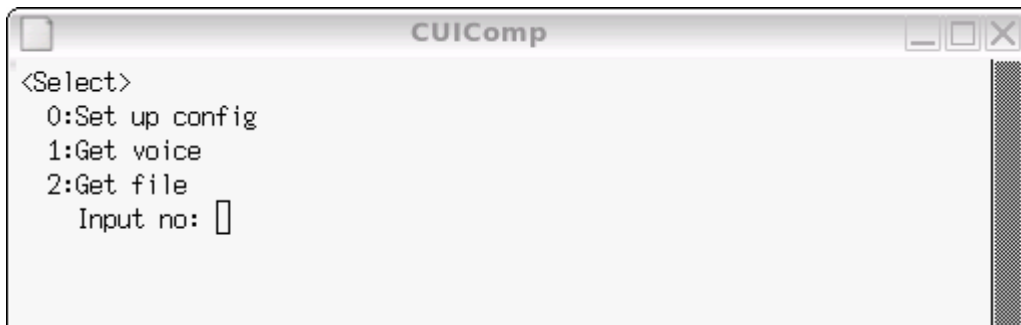
次に、ツールバーの All Active ボタン（下図参照）を押して、全ての RT コンポーネントをアクティブ状態にする（アクティブ状態になると、RT コンポーネントの色が、緑色（デフォルトの色設定）に変わる）。



以上の操作で、CUI 版話者識別システムが、実行可能な状態となる。

6.1.3. 操作

コンソールウィンドウ「CUIComp」にて、CUI 版話者識別システムの操作を行う。初期状態では、以下の表示になっている。



6.1.3.1 話者識別のための設定を行う

コンソールウィンドウに、'0' (Set up config) を入力して、リターンキーを押す。各項目の設定値を入力するように促されるので、下図のように値を入力して、リターンキーを押す(BetaRna-dir の値だけは、各自の値を入力すること。それ以外は、図と同じ設定でよい)。なお、これらのパラメタを変更した場合は、パラメタ変更前に行った学習と、不整合が生じるので注意を要する。

```

CUIComp
<Select>
  0:Set up config
  1:Get voice
  2:Get file
  Input no: 0
  Input window function id(1) -> 1
  Input lower limit freq(101.000000) -> 101
  Input upper limit freq(6100.000000) -> 6100
  Input devide count(12) -> 12
  Input BetaRna-dir : /home/rt1/MyData
  
```

なお、括弧で括られている値は、現在の設定値である。各設定項目の意味は、以下の通りである。

設定項目	値	説明
window function	1-10	パワースペクトル取得の前処理としての窓関数種別。 Hanning 法, Blackman 法, Sin カーブ等が 1~10 の数値に振り当てられている一般的には 1 の Hanning 法が適当と思われる。 (本項目の値は、音声特徴量抽出コンポーネントの設定ファイルである、「ps.ini」の WindowFunction に保存される。1-10 の各数値の説明については、『8.1 章 音声特徴量抽出コンポーネントの設定ファイル』の WindowFunction の項を参照のこと。)
lower limit freq	整数 (1~9999)	パワースペクトル使用下限周波数。 (本項目の値は、「ps.ini」の LowerFreq に保存される。)
upper limit freq	整数 (1~9999)	パワースペクトル使用上限周波数。 (本項目の値は、「ps.ini」の UpperFreq に保存される。)
devide count	整数 (1~20)	特徴量の個数。得られたパワースペクトルの使用周波数を、この数値で等分し、区間和を 0~1 に正規化したものを特徴量として使用することになる。 (本項目の値は、「ps.ini」の DivideNum に保存される。)

BetaRNA-dir	文字列	学習・推論コンポーネントが使用するディレクトリ。学習結果が記録されたファイル等が、保存される。
-------------	-----	---

最後に、確認の問い合わせがある。'y'を押すと、入力した設定値が有効となる（下図）。

```

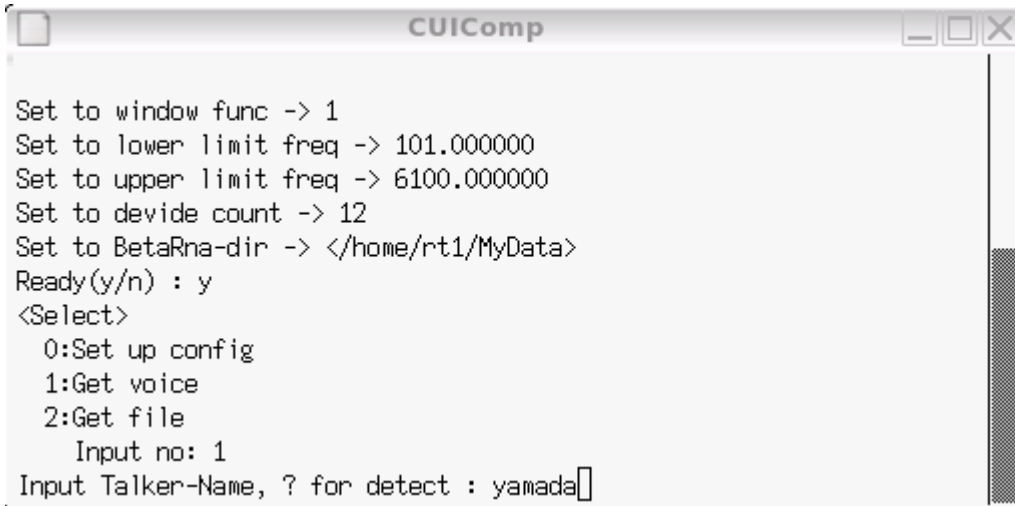
CUIComp
Input no: 0
Input window function id(1) -> 1
Input lower limit freq(101.000000) -> 101
Input upper limit freq(6100.000000) -> 6100
Input devide count(12) -> 12
Input BetaRna-dir : /home/rt1/MyData

Set to window func -> 1
Set to lower limit freq -> 101.000000
Set to upper limit freq -> 6100.000000
Set to devide count -> 12
Set to BetaRna-dir -> </home/rt1/MyData>
Ready(y/n) : y

```

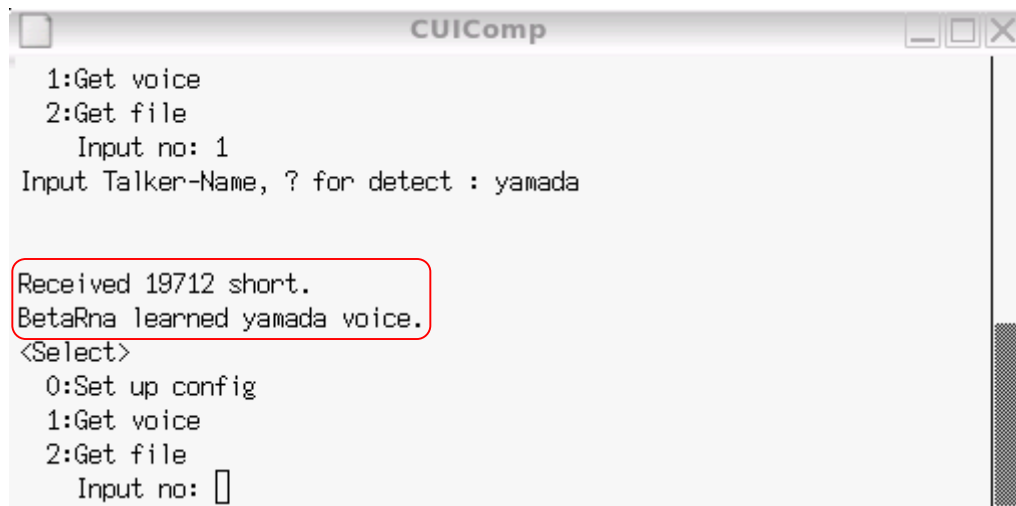
6.1.3.2 音声を学習させる（マイク）

マイクを使って、音声を学習させる場合、'1'を入力して、リターンキーを押す。話者が誰であるかを尋ねられるので、話者名を入力して、リターンを押す（下図）。



```
CUIComp
Set to window func -> 1
Set to lower limit freq -> 101.000000
Set to upper limit freq -> 6100.000000
Set to devide count -> 12
Set to BetaRna-dir -> </home/rt1/MyData>
Ready(y/n) : y
<Select>
  0:Set up config
  1:Get voice
  2:Get file
  Input no: 1
Input Talker-Name, ? for detect : yamada
```

話者名称を入力して、リターンキーを押すと、マイクによる音声入力が、可能な状態になる。この状態で、何か単語を発音する（例：「おはようございます」）。入力が成功すると、下図のような表示となる。

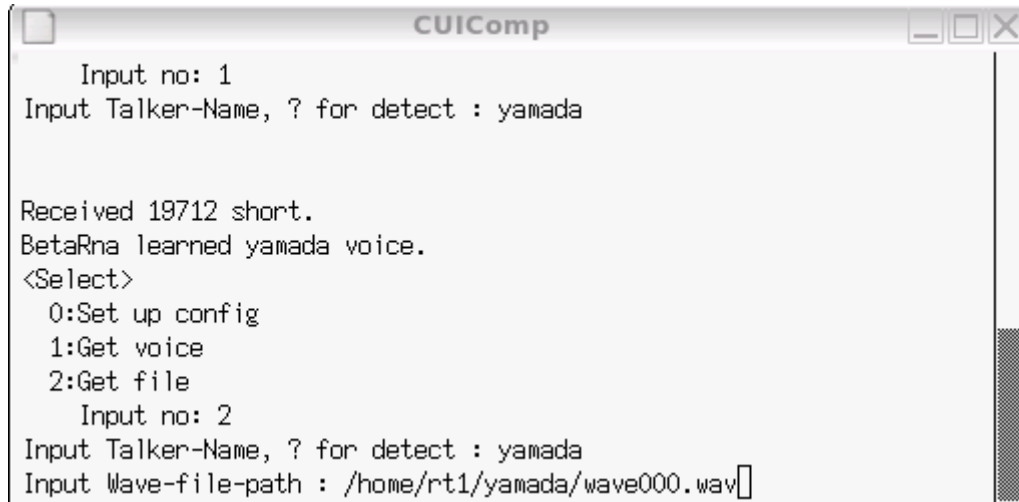


```
CUIComp
  1:Get voice
  2:Get file
  Input no: 1
Input Talker-Name, ? for detect : yamada

Received 19712 short.
BetaRna learned yamada voice.
<Select>
  0:Set up config
  1:Get voice
  2:Get file
  Input no: 
```


6.1.3.3 音声を学習させる（ファイル）

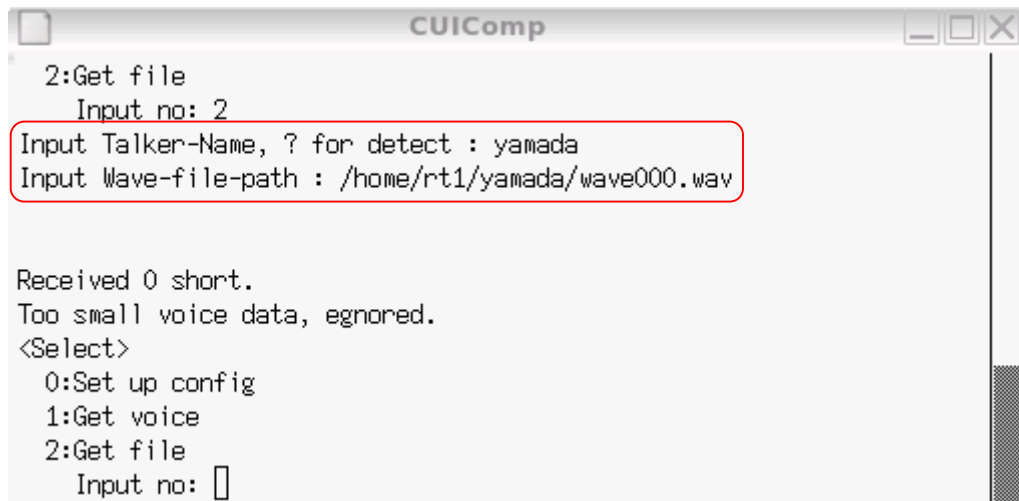
音声ファイル（wave ファイル）を使って、音声を学習させる場合、'2'を入力して、リターンキーを押す。話者が誰であるかを尋ねられるので、話者名を入力して、リターンを押す（下図）。次に、音声ファイルのパスを求められるので、音声ファイルのパスを入力して、リターンを押す（下図）。



```
CUIComp
Input no: 1
Input Talker-Name, ? for detect : yamada

Received 19712 short.
BetaRna learned yamada voice.
<Select>
  0:Set up config
  1:Get voice
  2:Get file
Input no: 2
Input Talker-Name, ? for detect : yamada
Input Wave-file-path : /home/rt1/yamada/wave000.wav
```

音声ファイルによる音声入力完了すると、下図のような表示となる。

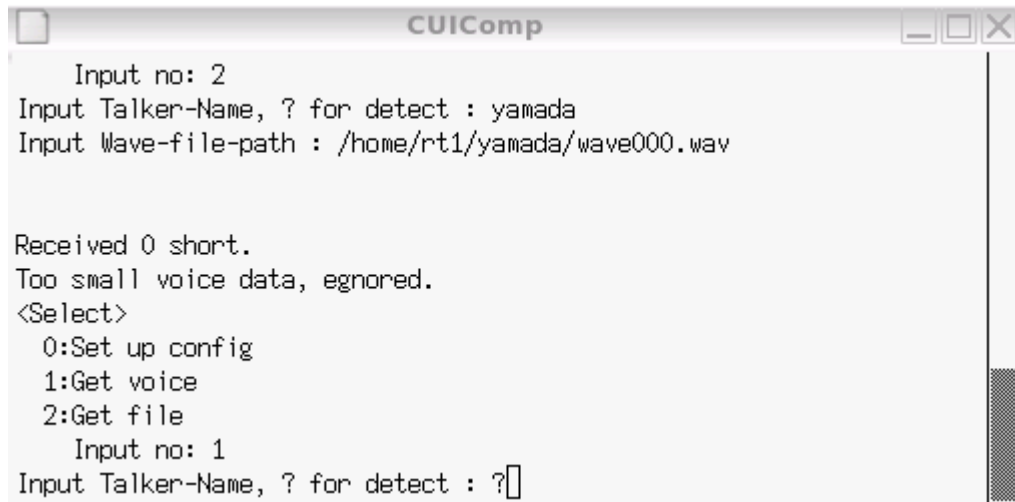


```
CUIComp
2:Get file
Input no: 2
Input Talker-Name, ? for detect : yamada
Input Wave-file-path : /home/rt1/yamada/wave000.wav

Received 0 short.
Too small voice data, egnored.
<Select>
  0:Set up config
  1:Get voice
  2:Get file
Input no: 
```

6.1.3.4 話者を推論させる (マイク)

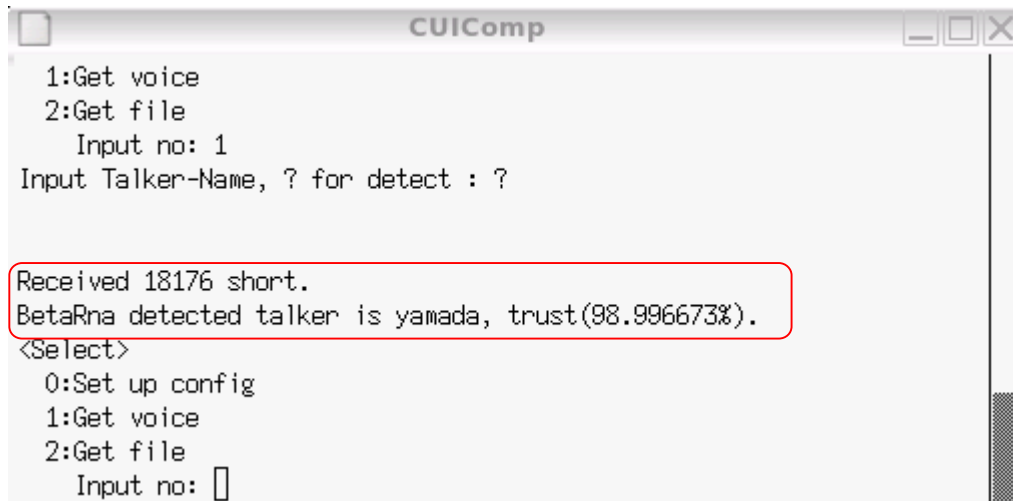
マイクに入力された音声で、「誰の音声であるか？」を推論させる場合、'1'を入力して、リターンキーを押す。話者が誰であるかを尋ねられるので、話者名を '?' として入力して、リターンを押す (下図)。



```
CUIComp
Input no: 2
Input Talker-Name, ? for detect : yamada
Input Wave-file-path : /home/rt1/yamada/wave000.wav

Received 0 short.
Too small voice data, egnored.
<Select>
  0:Set up config
  1:Get voice
  2:Get file
  Input no: 1
Input Talker-Name, ? for detect : ?
```

話者名 '?' 入力して、リターンキーを押すと、マイクによる音声入力が、可能な状態になる。この状態で、何か単語を発音する (例: 「おはようございます」)。入力が成功すると、下図のように、推論結果が表示される。



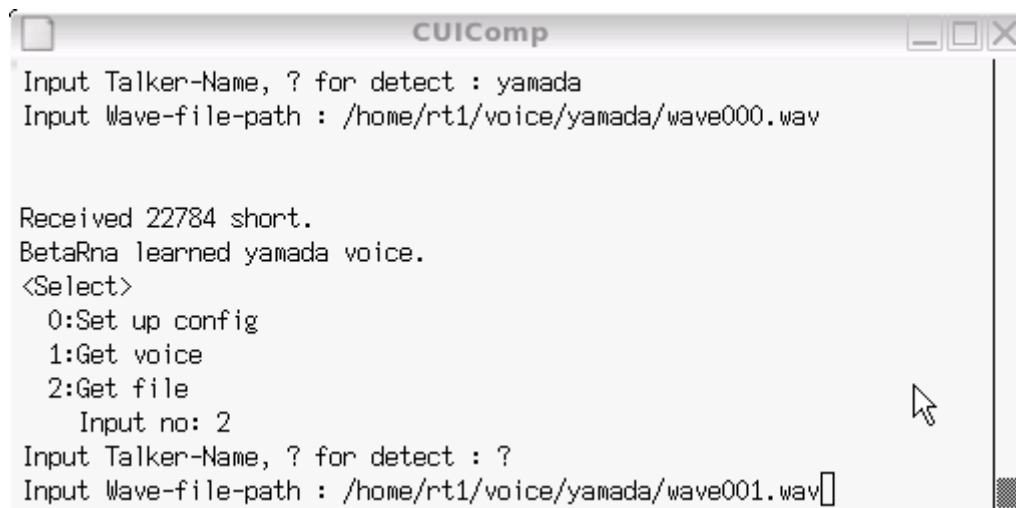
```
CUIComp
  1:Get voice
  2:Get file
  Input no: 1
Input Talker-Name, ? for detect : ?

Received 18176 short.
BetaRna detected talker is yamada, trust(98.996673%).
<Select>
  0:Set up config
  1:Get voice
  2:Get file
  Input no: 
```

この図の例では、入力された音声は、98.996673%の確率で、yamada の音声であるという推論結果を示している。

6.1.3.5 話者を推論させる (ファイル)

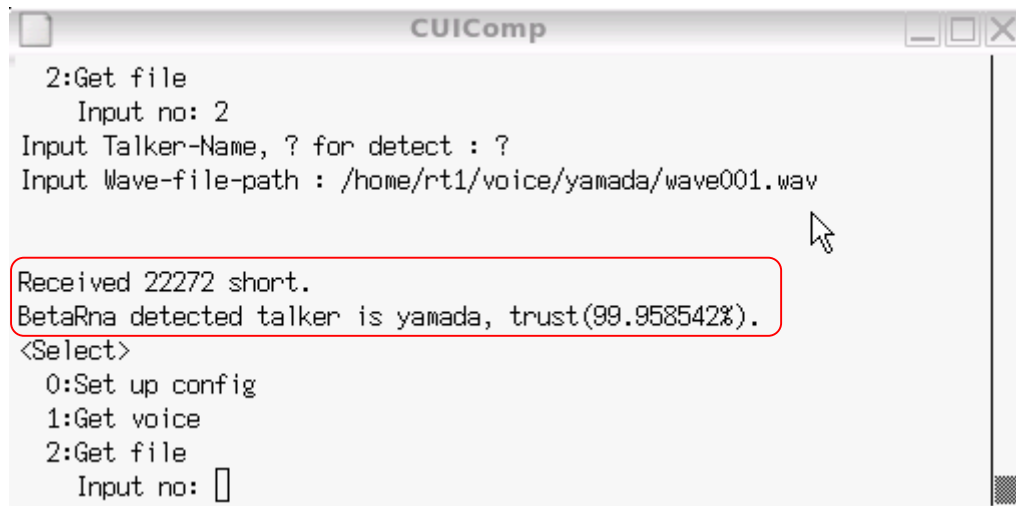
音声ファイル (wave ファイル) の音声で、「誰の音声であるか？」を推論させる場合、'2' を入力して、リターンキーを押す。話者が誰であるかを尋ねられるので、話者名 '?' を入力して、リターンを押す (下図)。次に、音声ファイルのパスを求められるので、音声ファイルのパスを入力して、リターンを押す (下図)。



```
CUIComp
Input Talker-Name, ? for detect : yamada
Input Wave-file-path : /home/rt1/voice/yamada/wave000.wav

Received 22784 short.
BetaRna learned yamada voice.
<Select>
  0:Set up config
  1:Get voice
  2:Get file
    Input no: 2
Input Talker-Name, ? for detect : ?
Input Wave-file-path : /home/rt1/voice/yamada/wave001.wav
```

音声ファイルによる音声入力完了すると、下図のように、推論結果が表示される。



```
CUIComp
  2:Get file
    Input no: 2
Input Talker-Name, ? for detect : ?
Input Wave-file-path : /home/rt1/voice/yamada/wave001.wav

Received 22272 short.
BetaRna detected talker is yamada, trust(99.958542%).
<Select>
  0:Set up config
  1:Get voice
  2:Get file
    Input no: 
```

この図の例では、入力された音声は、99.958542%の確率で、yamada の音声であるという推論結果を示している。

6.1.4. 終了

話者識別システムを終了するには、スクリプトファイル "インストールディレクトリ /run/stop.sh" を実行する。

6.2. GUI 版話者識別システム

6.2.1. 起動

スクリプトファイル "インストールディレクトリ/run/gui.sh" を実行する。これにより、話者識別システムを構成する RT コンポーネントが、全て起動する。画面には、5つのコンソールウィンドウと、GUI ウィンドウ (BetaRna voice detect demo) が表示される。この状態では、各 RT コンポーネントは、全てアイドル状態である。

6.2.2. システム構築

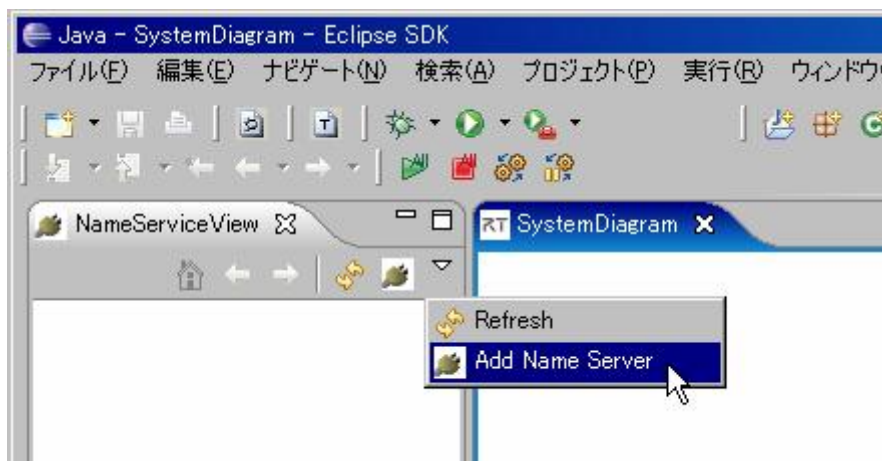
(4) RTCLink 起動する。

RTCLink の起動、操作方法は、以下のページを参照のこと。

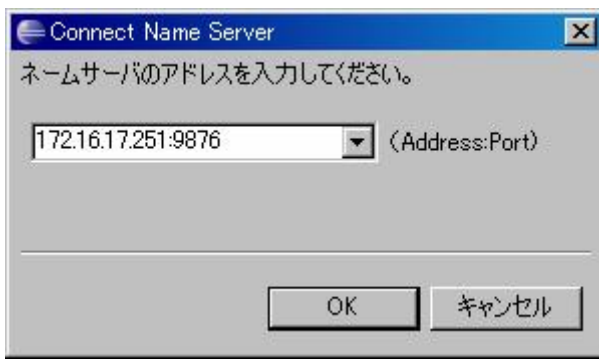
<http://www.is.aist.go.jp/rt/OpenRTM-aist/html/FrontPage.html>

(5) ネームサーバに接続する。

NameServiceView から、[AddNameServer]メニューを選択する (下図参照)。

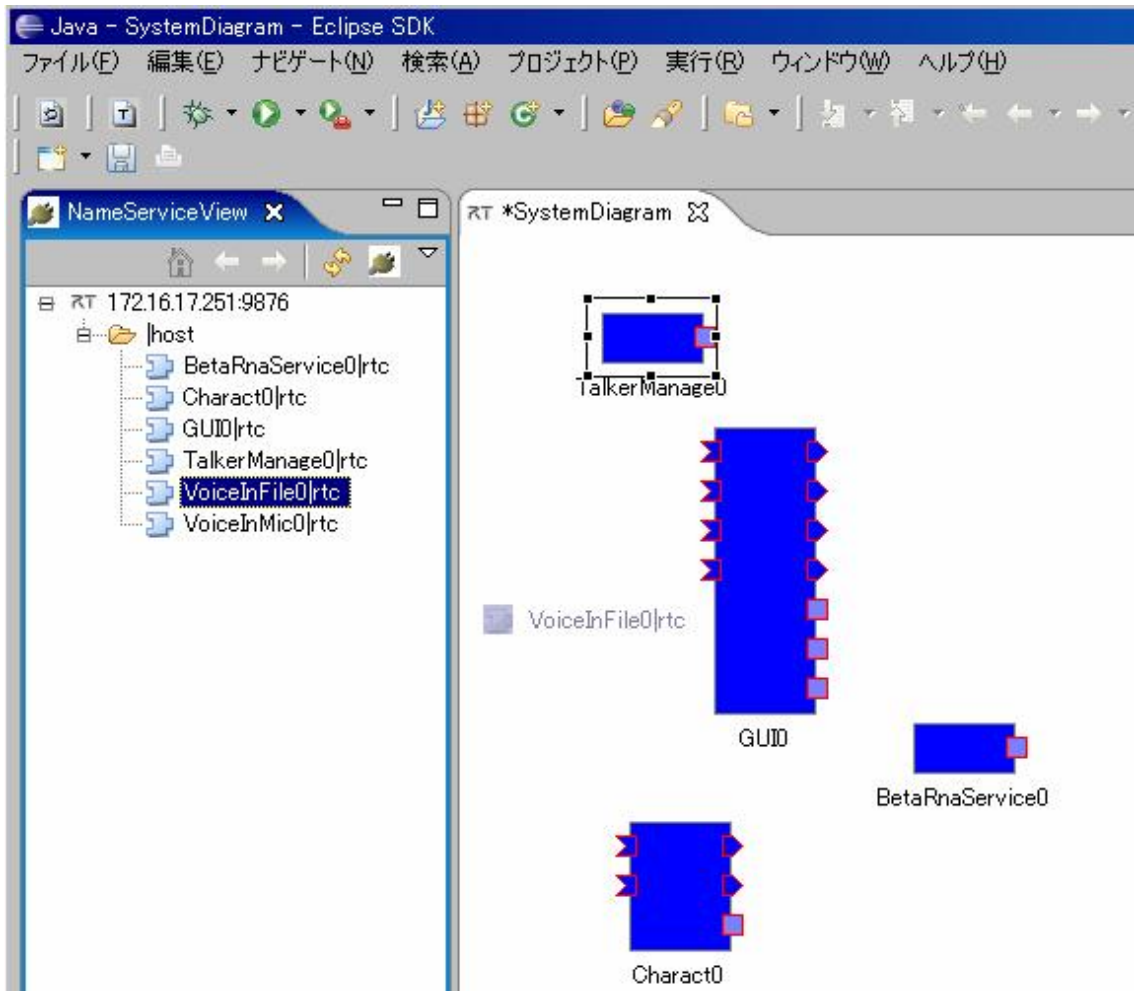


ConnectNameServer 画面で、ネームサーバを指定する (下図参照)。ネームサーバのアドレス、ポート番号は、設定ファイル "インストールディレクトリ/run rtc.conf" の項目 **corba.nameservers** で指定したものを記述すればよい。

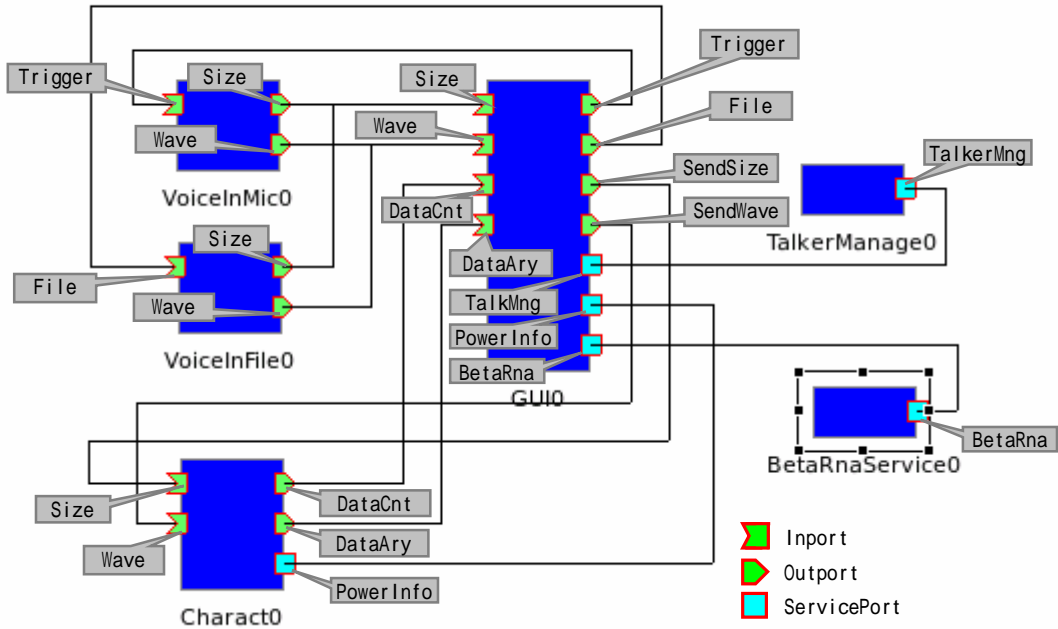


(6) RT コンポーネントの接続を行う

[ファイル]-[OpenNewSystemEditor]メニューを選択する。SystemDiagram ウィンドウが生成するので、そこに、NameServiceView で表示されている RT コンポーネント全てを、ドラッグアンドドロップ操作により、配置する。



次に、配置した RT コンポーネントのポート同士を接続する。ポートの接続は、下図の通りとする。



次に、ツールバーの All Active ボタン（下図参照）を押して、全ての RT コンポーネントをアクティブ状態にする（アクティブ状態になると、RT コンポーネントの色が、緑色（デフォルトの色設定）に変わる）。

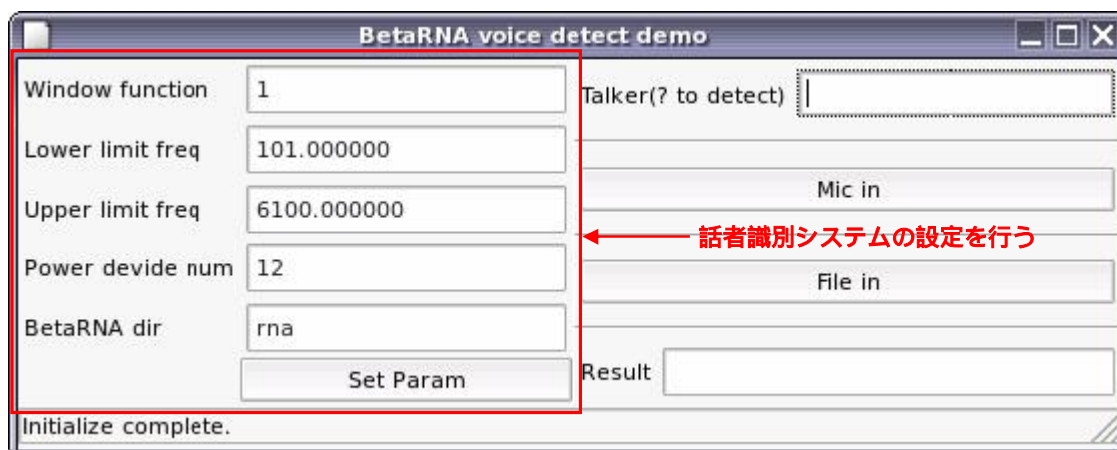


以上の操作で、GUI 版話者識別システムが、実行可能な状態となる。

6.2.3. 操作

6.2.3.1 話者識別のための設定を行う

話者識別システムの起動、構築が完了したら、話者識別システムの設定を行う。話者識別システムの設定は、GUI ウィンドウの左側で行う。



～ は、音声データの特徴量化に関する設定である。これらは、通常、上図の通りの値でよい。の BetaRNA dir のみ、各自の値を設定すること。なお、～ のパラメタを変更した場合は、パラメタ変更前に行った学習と、不整合が生じるので注意を要する。各項目の設定が終了したら、Set Param ボタンを押して、設定変更を反映させる。

なお、各設定項目の意味は、以下の通りである。

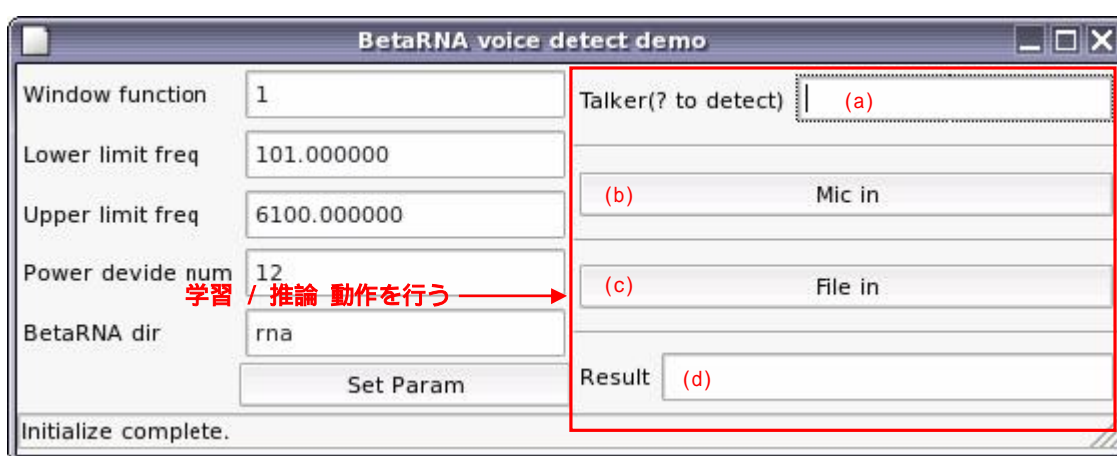
設定項目	値	説明
Window function	1-10	パワースペクトル取得の前処理としての窓関数種別。 Hanning 法, Blackman 法, Sin カーブ等が 1~10 の数値に振り当てられている一般的には 1 の Hanning 法が適当と思われる。 (本項目の値は、音声特徴量抽出コンポーネントの設定ファイルである、「ps.ini」の WindowFunction に保存される。1-10 の各数値の説明については、『8.1 章 音声特徴量抽出コンポーネントの設定ファイル』の WindowFunction の項を参照のこと。)
Lower limit freq	整数 (1~9999)	パワースペクトル使用下限周波数。 (本項目の値は、「ps.ini」の LowerFreq に保存される。)
Upper limit freq	整数 (1~9999)	パワースペクトル使用上限周波数。 (本項目の値は、「ps.ini」の UpperFreq に保存される。)
Power devide num	整数	特徴量の個数。得られたパワースペクトルの使用周波数を、この数値で

	(1~20)	等分し、区間和を 0~1 に正規化したものを特徴量として使用することになる。 (本項目の値は、「ps.ini」の DivideNum に保存される。)
BetaRNA dir	文字列	学習・推論コンポーネントが使用するディレクトリ。学習結果が記録されたファイル等が、保存される。

以上の操作で、話者識別システムを使用することができる。

6.2.3.2 学習 / 推論を行わせる

学習・推論を行わせる場合は、GUI ウィンドウの右側を使用する。



[学習を行わせる場合]

- (1) Talker 入力欄(a)に、話者名（英文字）を入力する。
- (2) 話者の音声データを入力する。
 1. 音声をマイク入力する場合は、Mic in ボタン(b)を押した後、マイクで音声を入力する。
 2. 音声ファイルのデータを使用する場合は、File in ボタン(c)を押す。File in ボタンを押すと、ファイル選択ウィンドウが表示される。ファイル選択ウィンドウ上で、使用するファイルを選択して、OK を押す。
- (3) 以上で、1 回の学習が完了する。

[推論を行わせる場合]

- (1) Talker 入力欄(a)に、' ? ' を入力する。
- (2) 推論対象の音声データを入力する。
 1. 音声をマイク入力する場合は、Mic in ボタン(b)を押した後、マイクで音声を入力する。
 2. 音声ファイルのデータを使用する場合は、File in ボタン(c)を押す。File in ボタン

を押すと、ファイル選択ウィンドウが表示される。ファイル選択ウィンドウ上で、使用するファイルを選択して、OK を押す。

- (3) Result 欄(d)に、推論結果（話者名）が表示される。なお、学習を一切行っていない状態で推論を行うとエラーとなる。

6.2.4. 終了

話者識別システムを終了するには、スクリプトファイル "インストールディレクトリ/run/stop.sh" を実行する。

7. コンポーネントのインターフェース仕様

7.1. インターフェース仕様

話者識別システムで用いる RT コンポーネントのインターフェースを、以下に示す。

7.1.1. 音声入力コンポーネント（マイク版）

機能： 入力開始トリガを受けて、マイクからの入力を Wave 形式で出力する

名称： VoiceInMic

ポート：

InPort

ポート名称	データ型	内容
Trigger	TimedString	入力開始トリガ

OutPort

ポート名称	データ型	内容
Size	TimedLong	音声データバイト数
Wave	TimedShortSeq	Wave データ

7.1.2. 音声入力コンポーネント（ファイル版）

機能： 入力ファイル名を受けて、ファイルのデータを Wave 形式で出力する

名称： VoiceInFile

ポート：

InPort

ポート名称	データ型	内容
File	TimedString	Wave ファイル名

OutPort

ポート名称	データ型	内容
Size	TimedLong	音声データバイト数
Wave	TimedShortSeq	Wave データ

7.1.3. 話者管理コンポーネント

機能： ServicePort により、話者名と識別 ID の関係を管理する。

名称： TalkerManage

ポート：

ServicePort (provider: port name=TalkerMng)

関数名称	戻り値	戻り値の内容	引数	引数の内容
GetID	long	識別 ID	string sName	話者名
GetName	string	話者名	long ID	識別 ID

7.1.4. 音声特徴量抽出コンポーネント

機能： Wave データを FFT 解析しパワースペクトルを求め、周波数帯別に平均した値（これを音声の特徴量とする）を出力する

名称： Charact

ポート：

InPort

ポート名称	データ型	内容
Size	TimedLong	音声データバイト数
Wave	TimedShortSeq	Wave データ

OutPort

ポート名称	データ型	内容
DataCnt	TimedShort	特徴量のデータ数
DataAry	TimedDoubleSeq	特徴量のデータ配列

ServicePort (provider: port name=PowerInfo)

関数名称	戻り値	戻り値の内容	引数	引数の内容
SetParam	short	処理結果	short iWndFunc	窓関数
			double dLowerLimit	パワースペクトル使用下限周波数
			double dUpperLimit	パワースペクトル使用上限周波数
			short iOutputCount	特徴量の個数
GetParam	short	処理結果	short& iWndFunc	窓関数
			double& dLowerLimit	パワースペクトル使用下限周波数
			double& dUpperLimit	パワースペクトル使用上限周波数
			short& iOutputCoun	特徴量の個数

7.1.5. 学習・推論コンポーネント（データポート版）

機能：学習・推論を行う。

名称：BetaRnaPort

ポート：

InPort

ポート名称	データ型	内容
ID	TimedShort	話者 ID (-1 が入力されると推論動作を行う)
ParamDir	TimedString	パラメタディレクトリパス
DataCount	TimedShort	特徴量のデータ数
DataAry	TimedDoubleSeq	特徴量のデータ配列

OutPort

ポート名称	データ型	内容
Result	TimedShort	推論結果の話者 ID
Trusty	TimedDouble	推論結果の合致度 (推論結果がどの位の確率で正しいかを表す指標)

7.1.6. 学習・推論コンポーネント（サービスポート版）

機能：ServicePort によりデータを授受して学習 / 推論を行う。

名称：BetaRnaService

ポート：

ServicePort (provider: port name=BetaRna)

関数名	戻り値	戻り値の内容	引数	引数の内容
Init	short	処理結果	string sRnaDir short iInputCh	パラメタディレクトリ 特徴量の個数
Exec	short	処理結果	short iTalkerID DArray(*) aInputData short& iResultID double& dTrusty	話者 ID 特徴量データ配列 推論結果の話者 ID 確信度

(*) DArray : sequence<double>のこと

7.1.7. ユーザーインターフェースコンポーネント（CUI版）

機能：ユーザに、話者識別システムのコンソールインターフェースを提供する。

また、他のコンポーネントの上位のコンポーネントとして機能する。

名称： CUI

ポート：

InPort

名称	データ型	内容
Size	TimedLong	音声データバイト数
Wave	TimedShortSeq	Wave データ
ResultID	TimedShort	推論結果の話者 ID
Trusty	TimedDouble	推論結果の合致度 (推論結果がどの位の確率で正しいかを表す指標)

OutPort

名称	データ型	内容
Trigger	TimedString	入力開始トリガ
File	TimedString	Wave ファイル名
SendSize	TimedLong	音声データバイト数
SendWave	TimedShortSeq	Wave データ
ID	TimedShort	話者 ID (推論時は、話者 ID を、 -1 とする)
ParamDir	TimedString	学習・推論コンポーネントが使用するディレクトリパス。学習結果が記録されたファイル等が、保存される。

ServicePort (client : port name = TalkMgn、 PowerInfo、 BetaRna)

関数名称	関数を実行するコンポーネント	サービスポート
GetID	TalkManager コンポーネント	TalkMgn ポート
GetName	TalkManager コンポーネント	TalkMgn ポート
SetParam	Charact コンポーネント	PowerInfo ポート
GetParam	Charact コンポーネント	PowerInfo ポート

7.1.8. ユーザインターフェースコンポーネント (GUI 版)

機能： ユーザに、話者識別システムの、グラフィカルなインターフェースを提供する。
また、他のコンポーネントの上位のコンポーネントとして機能する。

名称： GUI

ポート：

InPort

名称	データ型	内容
Size	TimedLong	音声データバイト数

Wave	TimedShortSeq	Wave データ
DataCnt	TimedShort	特徴量のデータ数
DataAry	TimedDoubleSeq	特徴量のデータ配列

OutPort

名称	データ型	内容
Trigger	TimedString	入力開始トリガ
File	TimedString	Wave ファイル名
SendSize	TimedLong	音声データバイト数
SendWave	TimedShortSeq	Wave データ

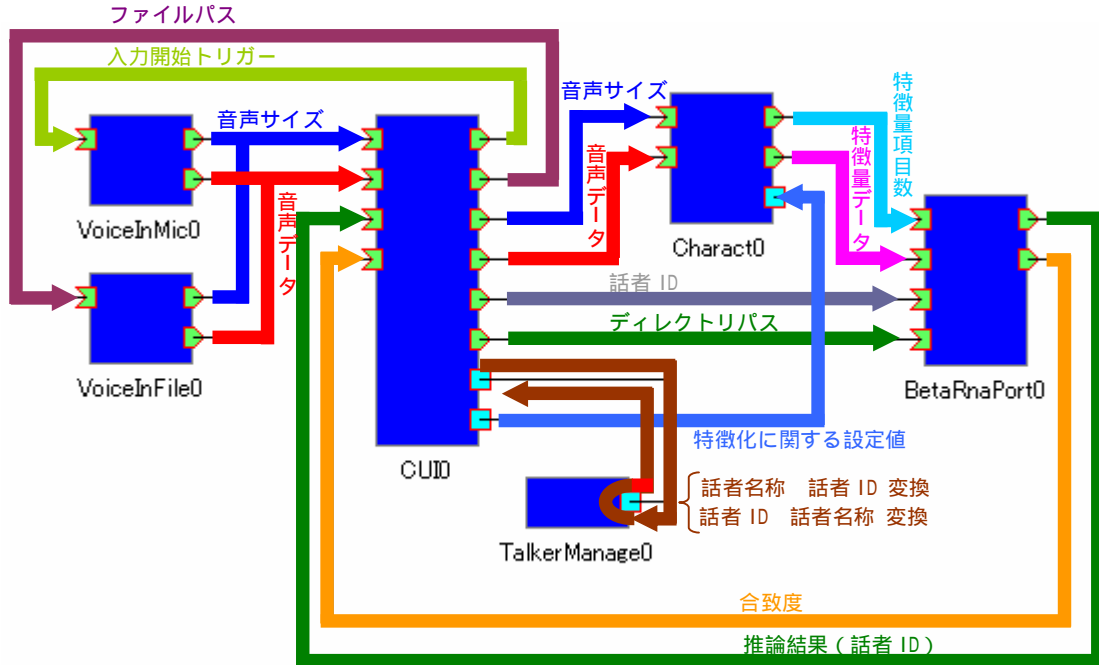
ServicePort (client : port name = TalkMgn、PowerInfo、BetaRna)

関数名	関数を実行するコンポーネント	サービスポート
GetID	TalkManager コンポーネント	TalkMgn ポート
GetName	TalkManager コンポーネント	TalkMgn ポート
SetParam	Charact コンポーネント	PowerInfo ポート
GetParam	Charact コンポーネント	PowerInfo ポート
Init	BetaRna コンポーネント	BetaRna ポート
Exec	BetaRna コンポーネント	BetaRna ポート

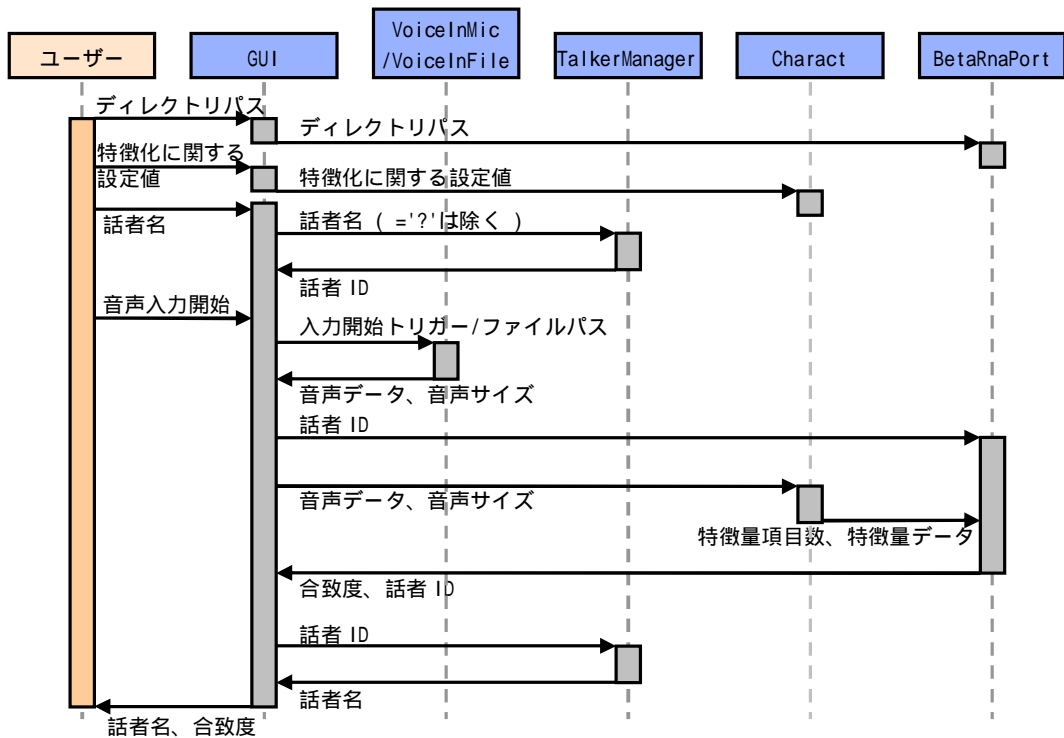
7.2. コンポーネント間シーケンス

7.2.1. CUI 版話者識別システム

コンポーネント間のデータ送受信の概略図を、下に示す。

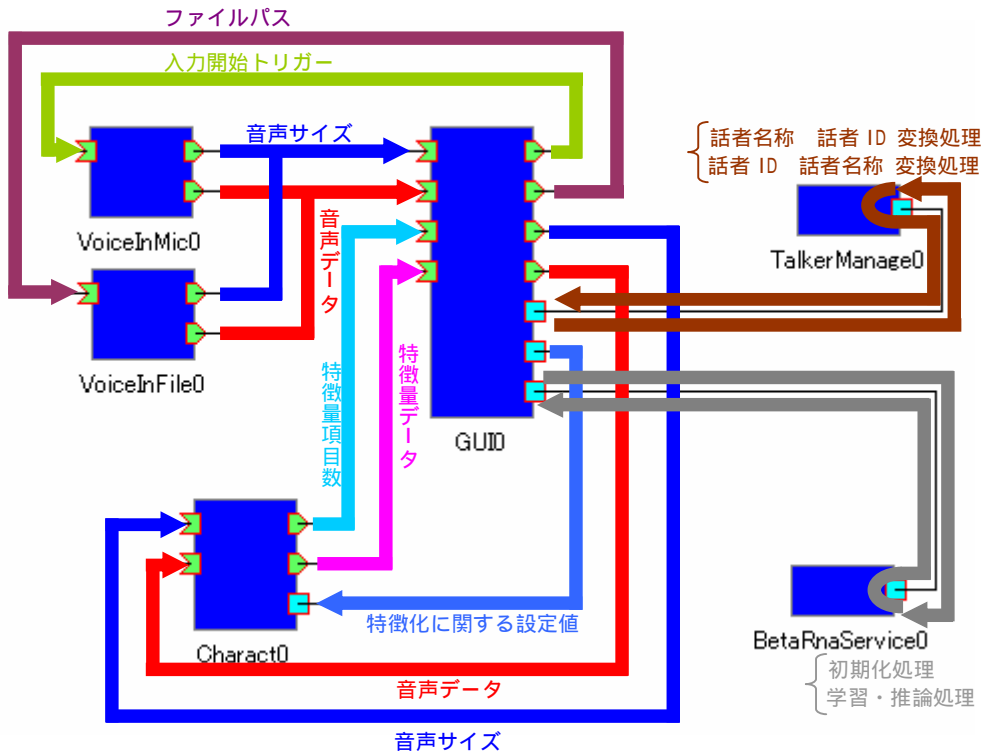


これらのシーケンス図を、以下に示す。

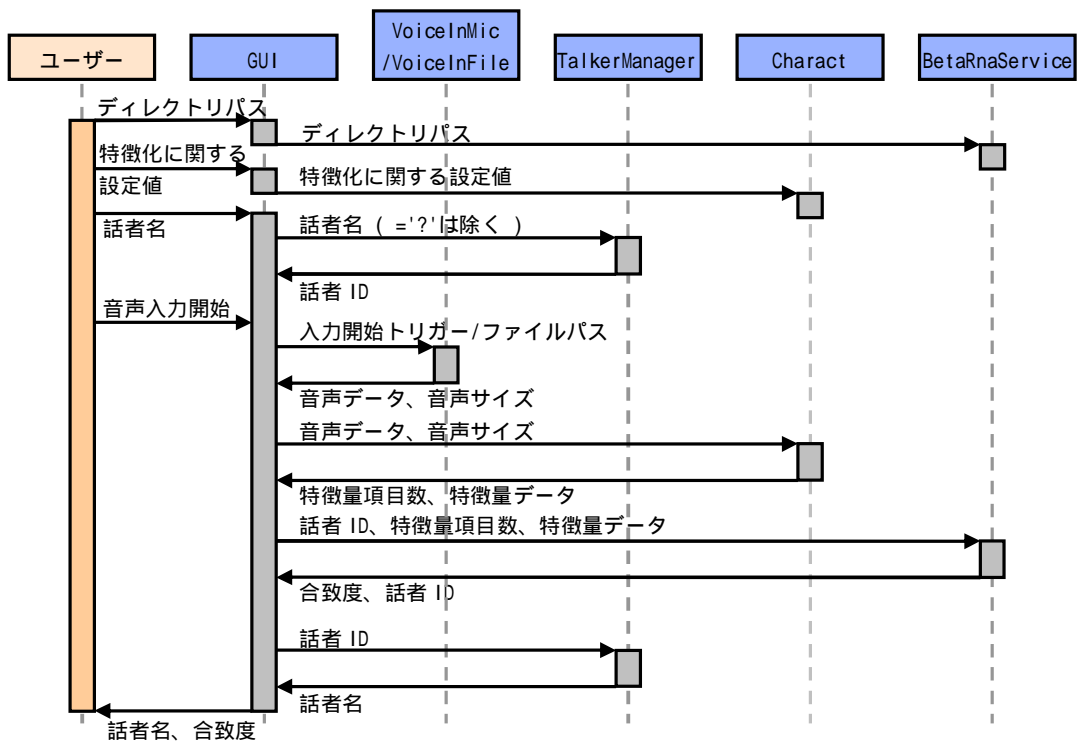


7.2.2. GUI 版話者識別システム

コンポーネント間のデータ送受信の概略図を、下に示す。



これらのシーケンス図を、以下に示す。



8. コンポーネント設定ファイル仕様

話者識別システムを構成する RT コンポーネントの中で、設定ファイルが必要なコンポーネントは、VliceInMic コンポーネント、Charact コンポーネント、BetaRnaService コンポーネントの3つである。本章では、各コンポーネントの設定ファイルの内容について、説明する。なお、通常、これらの設定ファイルを直接編集する必要はない(例外として、インストール時に、一度のみ編集する必要がある。これについては、『6.1. インストール時(初回起動前)の設定』を参照すること)。

8.1. 音声特徴量抽出コンポーネントの設定ファイル

設定ファイル : ps.ini

設置場所 : Charact コンポーネントと同一階層

設定項目 : 下表の通り

設定項目	値	説明
WindowFunction	0	square window (矩形窓)
	1	hanning window (ハニング窓)
	2	hamming window (ハミング窓)
	3	blackman window (ブラックマン窓)
	4	blackman harris window (ブラックマン ハリス窓)
	5	bartlett window (バートレット窓)
	6	parzen window (パルザン窓)
	7	welch window (ウェルチ窓)
	8	riesz window
	9	sin window (サイン窓)
	10	triangular window (三角窓)
LowFreq	整数 (1 ~ 9999)	パワースペクトルの使用下限周波数
UpperFreq	整数 (1 ~ 9999)	パワースペクトルの使用上限周波数
DivideNum	整数 (1 ~ 20)	特徴量の個数。得られたパワースペクトルの使用周波数を、この数値で等分して、区間和を 0 以上 1 以下に正規化したものを、特徴量として使用する。
SamplingFreq	整数 (1 ~ 99999)	音声のサンプリングレート。音声が入力されるマシンのサウンドデバイスの設定に依存する。(マイク音声入力コンポーネントの設定ファイルの SamplingRate で指定された値と同じ値でなければならない)

8.2. 音声入力コンポーネント（マイク版）の設定ファイル

設定ファイル：jr.ini

設置場所：VoiceInMic コンポーネントと同一階層

設定項目：下表の通り

設定項目	値	説明
PortNo	整数 (1～65535)	音声入力ツール「Julius adintool」が使用するポート番号
AdintoolPath	文字列	音声入力ツール「Julius adintool」へのフルパス
SamplingRate	整数 (1～99999)	取得する音声のサンプリングレート。音声が入力されるマシンのサウンドデバイスの設定に依存する。(音声特徴量抽出コンポーネントの設定ファイルの SamplingFreq で指定された値と同じ値でなければならない)
Select Interval	整数 (1～1000)	音声入力ツール「Julius adintool」とソケット通信を行う際の、データ受信時のセレクトタイムアウト。内部的な値なので、特に変更する必要はない。400～500 が妥当。
SegmentUse	0	有音判定を行わない。無音データも含めて、RecordMaxTime で指定された秒数分の音声データを取得する。
	1	有音判定を行う。一連の有音状態のときのみ、音声取得を行う（無音データは取得しない）。なお、有音状態が、RecordMaxTime で指定された秒数に達した場合は、音声取得を終了する。
RecordMaxTime	整数 (1～30)	取得最大時間。秒数で指定する。
FileOut	0	取得した音声データを、ファイルに保存しない。
	1	取得した音声データを、ファイルに保存する。 ファイル名称は、次の通りである。 'カレントディレクトリ'/'話者名'/wave'連番'.wav

8.3. 学習・推論コンポーネントの設定ファイル

設定ファイル : Rs1.ila

設置場所 : Init 関数の第一引数 sRnaDir で指定されたフォルダ

設定項目 : 省略(特に編集する必要はない。独自のチューニングを行いたいときは、『rs1 データファイル書式仕様書.pdf』を参照のこと。)

9. その他

9.1. ライセンス等

学習・推論コンポーネントの著作権は、-RNA ライブラリを除き、独立行政法人産業技術総合研究所に帰属する。

-RNA ライブラリの著作権は、株式会社アドイン研究所に帰属する。

-RNA ライブラリは、株式会社アドイン研究所の商品であるが、非商用利用に限っては保守・サポートなしを条件に無償で使用してよい。

9.2. 連絡先

株式会社 アドイン研究所

〒102-0094 東京都千代田区紀尾井町 3-6 秀和紀尾井町パークビル 1F

TEL: 03-3511-2215 FAX: 03-3511-3078

E-mail: rt_brna@adin.co.jp

URL: <http://www.adin.co.jp>

- 以上 -