

コミュニケーション機能共通インタフェース仕様書 (第1.0 版)

NEDO 次世代ロボット知能化技術開発プロジェクト

2012 年 2 月 24 日



【改版履歴】

日付	版番号	改版ページ	改版内容
2012. 2. 24	1. 0	全ページ	新規作成

【本書の利用にあたって】

本書は、クリエイティブ・コモンズ 表示 2.1 ライセンスの下に提供される。

(<http://creativecommons.org/licenses/by-sa/2.1/jp/>)



【本書の策定メンバー】

(敬称略、五十音順)

小笠原哲也 (東京大学大学院 情報理工学系研究科 知能機械情報学専攻)

高野陽介 (日本電気株式会社)

中本啓之 (株式会社セック 開発本部 第四開発部)

二宮恒樹 (富士ソフト株式会社 ロボット事業グループ 商品開発ユニット)

原功 (独立行政法人産業技術総合研究所)

松坂要佐 (独立行政法人産業技術総合研究所)

(所属は2012年2月24日現在)

目次

1	はじめに.....	1
1.1	対象機能の概要.....	1
1.2	標準システム構成.....	2
2	本書を読む上での注意.....	3
2.1	基本方針.....	3
2.2	フォーマットと表現方法.....	3
2.2.1	型定義.....	3
2.3	本仕様書における前提条件.....	4
2.3.1	音声データ.....	4
2.3.2	音声認識文法フォーマット.....	5
2.3.3	音声合成フォーマット.....	7
3	名前空間定義.....	8
4	データ型定義.....	8
4.1	標準型.....	8
4.1.1	RTC::TimedOctetSeq.....	8
4.1.2	RTC::TimedBoolean.....	8
4.1.3	RTC::TimedString.....	8
4.2	型宣言.....	9
4.3	コミュニケーション機能用.....	9
5	共通インタフェース定義.....	10
5.1	データポート.....	10
5.1.1	音声データインタフェース.....	10
5.1.2	Voice Activity インタフェース.....	10
5.1.3	認識ステータスインタフェース.....	10
5.1.4	認識結果インタフェース.....	11
5.2	サービスポート.....	12
6	共通インタフェースを利用したシステム構成例.....	13
6.1	対話制御コンポーネント群 OpenHRI.....	13
7	CORBA IDL.....	14
8	参考文献.....	15

表目次

表 4.1 RTC::TimedOctetSeq	8
表 4.2 RTC::TimedBoolean.....	8
表 4.3 RTC::TimedString	8
表 5.1 音声認識状態	10
表 5.2 XML タグ定義	12

図目次

図 1.1 コミュニケーション機能共通インタフェースの使用シーン例	1
図 1.2 コミュニケーション機能共通インタフェースを使用したシステム例	2
図 2.1 音声データのバイト配列	4
図 5.1 音声データインタフェース	10
図 5.2 Voice Activity インタフェース.....	10
図 5.3 認識ステータスインタフェース	10
図 6.1 コミュニケーション機能コンポーネント適用例	13
図 6.2 対話制御コンポーネント群 OpenHRI	13

1 はじめに

近年、ロボットの開発を効率化するためにコンポーネントベースのミドルウェア開発が盛んになっている。コンポーネントベースのミドルウェア開発において、インタフェースの共通化は、コンポーネントの相互接続性や相互運用性を確保するうえで非常に重要である。このような背景に基づき、本書では、音声を用いたロボットとのヒューマンコミュニケーション機能に関わる部分のインタフェースの共通仕様を定義する。

本共通インタフェースを規定することにより、ロボットへの音声入力、音声出力に関係する部分を共通化することができるため、ロボットとの対話制御が容易になるといったメリットが期待できる。

1.1 対象機能の概要

本仕様書では、ロボットシステムに対話機能を付加するための音声認識、音声合成モジュールの共通インタフェースを規定している。

コミュニケーション機能共通インタフェースを実装した RT コンポーネントの使用シーンの一例を以下に示す。

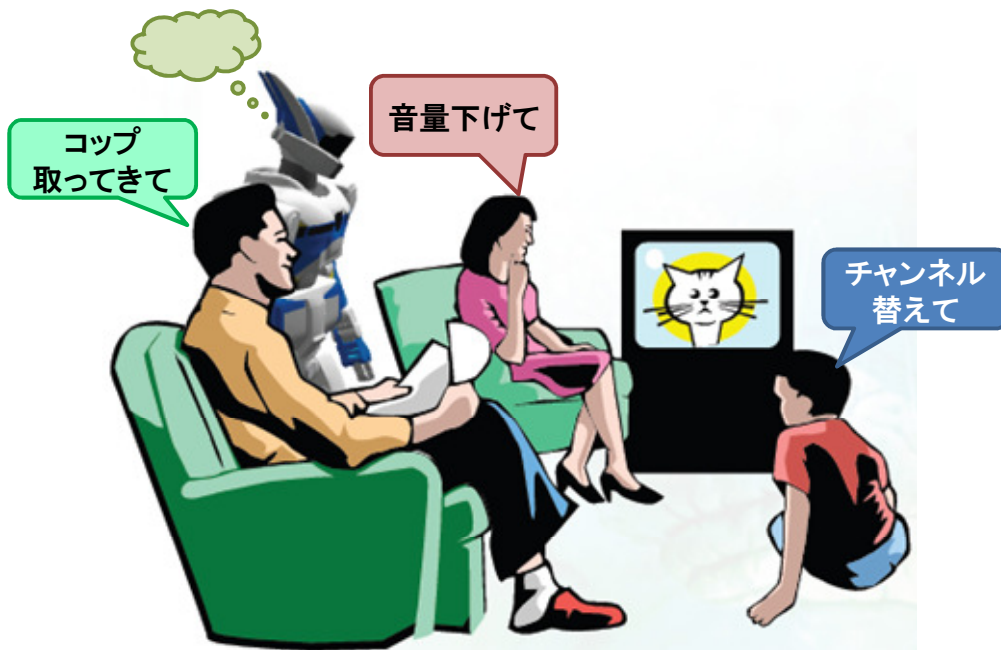


図 1.1 コミュニケーション機能共通インタフェースの使用シーン例

上記の例は、人間の生活空間内で活動するサービスロボットに対して音声で指示を与えている例である。このような対人サービスロボットの場合、利用される状況や、期待される動きは多岐にわたるため、利用者に特殊な操作方法を訓練したり、特殊な専門知識を学習させたりするのは、利便性の面からも無理がある。このため、人にとって自然な音声対話によって指示を与えられることが望ましい。また音声による指示を実現した場合でも、複数のユーザが同時に発話する場合や、周辺環境に騒音がある場合、更には同じ語句でも状況によって、期待される動作が異なる場合などもあるため、ロボット側に状況に応じたより知的な意図理解が求められる。このため、音声認識機能、音声合成機能などを有した対話エンジンの役割がより重要となってくる。

1.2 標準システム構成

コミュニケーション機能共通インタフェースを利用した標準的なシステム構成例を以下に示す。

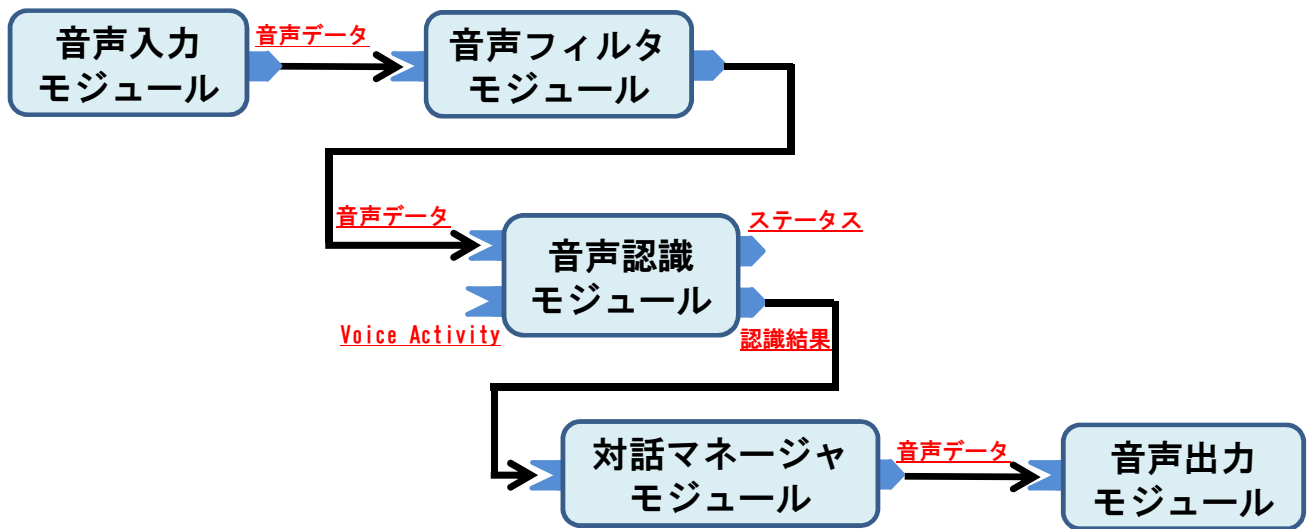


図 1.2 コミュニケーション機能共通インタフェースを使用したシステム例

本仕様書中では、上図中の各モジュールの入出力インタフェースを規定している。

「音声入力モジュール」「音声出力モジュール」は、ハードウェアの音声入出力デバイスにアクセスし、OpenRTM形式のデータストリームを入出力するモジュールである。Linux の alsa などの標準インタフェースや、マルチチャネル A/D, D/A コンバータなどの特殊デバイスに対応した実装が想定される。

「音声フィルタモジュール」は、OpenRTM 形式のデータストリームを入力とし、何らかの処理を加えた後、同形式のデータストリームを出力するモジュールである。音声認識の前処理に必要な、強調処理や、エコーキャンセル処理などに対応した実装が想定される。

「音声認識モジュール」は、OpenRTM形式のデータストリームを入力とし、認識されたテキストを出力するモジュールである。音声認識ソフトウェアには、認識アルゴリズムや、認識対象(言語の差など)によって様々な種類があるが、本書の共通規格に従うことで、それらを差し替えて利用出来るようになる。

「対話マネージャモジュール」は、音声認識モジュールから出力された認識テキストを入力とし、ユーザとの対話に関する処理を行った後、他のモジュールへのコマンドを出力するモジュールである。対話マネージャには、アルゴリズムや対象タスクによって様々な種類があるが、本書の共通規格に従うことで、それらを差し替えて利用できるようになる。

2 本書を読む上での注意

2.1 基本方針

インタフェース仕様の共通化は、仕様に合致しないコンポーネントを排除するため、時に開発内容を制限してしまうこともある。本仕様では、そのような制限を低減するために、以下のような方針で共通インタフェース仕様を定義する。

- 最低限のインタフェース仕様の定義:コンポーネントを相互接続・相互運用するために必要な最低限のインタフェース仕様のみを定義する。開発の制約となる仕様は最低限にとどめ、その他の部分は開発者が自由に拡張することができるようにする。
- 任意の機能の定義:いくつかの機能については実装を任意とする。実装された場合は、本書に書かれた仕様に準拠することを要求するが、実装をするかどうかは任意であり、それを実装していなかったからといって共通インタフェース仕様から外れるものとはしない。

2.2 フォーマットと表現方法

2.2.1 型定義

本仕様書では、型定義を次の表形式を用いて記述する。

表 XX <型名>

属性		
<要素名>	<要素型>	<内容>
...

2.3 本仕様書における前提条件

2.3.1 音声データ

音声データのバイト配列は以下の規則に従うものとする。

- 各サンプルを N 個、時間の遅いものから早いもの(最近のもの)に順列する。N は任意であり、N より長い音声ストリームを送りたい場合は、ストリームを複数のデータに分割することが出来る。
- 各サンプルには、各チャンネルが順列される。
- 各サンプルの各チャンネルの中には、音声データのバイト配列が順列される。バイト配列は、下位バイトから上位バイトに順列する。

上記規則に従った音声データの配列の例を以下に示す。

• 1チャンネル、8bitの 音声 (1サンプル 1Byte=Octet)

S1	S2	S3	S4											SN
----	----	----	----	-----	-----	-----	--	--	--	--	--	--	--	--	--	--	----

• 1チャンネル、16bitの 音声 (1サンプル 2Byte=Octet)

S1	S1	S2	S2										SN	SN
[1]	[2]	[1]	[2]													[1]	[2]

• 2チャンネル、16bitの 音声 (1サンプル 4Byte=Octet)

S1	S1	S1	S1									SN	SN	SN	SN
[1]	[2]	[1]	[2]												[1]	[2]	[1]	[2]
c1	c1	c2	c2												c1	c1	c2	c2

※Nは任意

(S*は各サンプル、[*]はバイト、c*は各チャンネルを意味する)

図 2.1 音声データのバイト配列

2.3.2 音声認識文法フォーマット

音声認識文法フォーマットは、W3C-Speech Recognition Grammar Specification (<http://www.w3.org/TR/speech-grammar/> §)に準拠する。

音声認識モジュールは、W3C-Speech Recognition Grammar Specification の定めるフォーマットのうち、XML 形式か ABNF 形式のいずれか、または両方を解釈出来る必要がある。

以下に XML 形式の認識文法の例を示す。

```
<?xml version="1.0" encoding="UTF-8" ?>
<grammar xmlns="http://www.w3.org/2001/06/grammar"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/06/grammar
    http://www.w3.org/TR/speech-grammar/grammar.xsd"
  xml:lang="jp" version="1.0" mode="voice" root="command">
  <rule id="command">
    <one-of>
      <item><ruleref uri="#greeting"/></item>
      <item><ruleref uri="#control"/></item>
    </one-of>
  </rule>
  <rule id="greeting">
    <one-of>
      <item>おはよう</item>
      <item>こんにちは</item>
      <item>こんばんは</item>
    </one-of>
  </rule>
  <rule id="control">
    <one-of>
      <item>前進</item>
      <item>バック</item>
    </one-of>
    <item repeat="0-1">
      <ruleref uri="#garbage"/>
    </item>
  </rule>
  <rule id="garbage">
    <one-of>
      <item>して</item>
      <item>してください</item>
    </one-of>
  </rule>
</grammar>
```

以下は ABNF 形式の認識文法の例である。

```
#ABNF 1.0;
language jp;
mode voice;
root $command;
$command = $greeting | $control ;
$greeting = (おはよう | こんにちは | こんにちは);
$control = (前進 | バック) [$garbage] ;
$garbage = (して | してください);
```

音声認識辞書フォーマットは、W3C-Pronunciation Lexicon Specification (<http://www.w3.org/TR/pronunciation-lexicon/> §)に準拠する。以下にその例を示す。

```
<?xml version="1.0" encoding="UTF-8"?>
<lexicon version="1.0"
  xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
    http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
  alphabet="kana" xml:lang="jp">
  <lexeme>
    <grapheme>おはよう</grapheme>
    <phoneme>{{KANA|おはよう}}</phoneme>
  </lexeme>
  <lexeme>
    <grapheme>こんにちは</grapheme>
    <phoneme>{{KANA|こんにちわ}}</phoneme>
  </lexeme>
  <lexeme>
    <grapheme>こんにちは</grapheme>
    <phoneme>{{KANA|こんばんわ}}</phoneme>
  </lexeme>
  <lexeme>
    <grapheme>前進</grapheme>
    <phoneme>{{KANA|ぜんしん}}</phoneme>
  </lexeme>
  <lexeme>
    <grapheme>バック</grapheme>
    <phoneme>{{KANA|ぱっく}}</phoneme>
  </lexeme>
  <lexeme>
    <grapheme>して</grapheme>
    <phoneme>{{KANA|して}}</phoneme>
  </lexeme>
  <lexeme>
    <grapheme>してください</grapheme>
    <phoneme>{{KANA|してください}}</phoneme>
  </lexeme>
</lexicon>
```

W3C-Pronunciation Lexicon Specification では、単語の発話を IPA 形式の音素記号によって記述できるようにすることが規格に定められているが、それ以外の任意の音素記号を用いて記述できるように拡張することが認められている。日本語などの場合は上記の例のように、カナ表記を使って記述可能にすると実用的であろう。

2.3.3 音声合成フォーマット

音声合成フォーマットは、W3C-Speech Synthesis Markup Language (<http://www.w3.org/TR/speech-synthesis/> §) に準拠する。以下に例を示す。

```
<?xml version="1.0" encoding="UTF-8"?>
<speak version="1.0" xmlns=http://www.w3.org/2001/10/synthesis
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.w3.org/2001/10/synthesis
        http://www.w3.org/TR/speech-synthesis/synthesis.xsd"
    xml:lang="ja">
    <p>日本語による音声合成フォーマットの例です。</p>
</speak>
```

3 名前空間定義

コミュニケーション機能共通インタフェースでは、固有の名前空間は定義していない。
名前空間の定義については、検討課題として継続して検討する。

4 データ型定義

コミュニケーション機能共通インタフェースで使用するデータ型を以下に示す。

4.1 標準型

4.1.1 RTC::TimedOctetSeq

時刻情報付きのバイナリデータを格納するための型。OpenRTM-aist の標準型として BasicDataType.idl 内で定義されている。

表 4.1 RTC::TimedOctetSeq

属性		
tm	RTC::Time	時刻情報
data	sequence<octet>	バイナリデータ列

4.1.2 RTC::TimedBoolean

時刻情報付きのブール値を格納するための型。OpenRTM-aist の標準型として BasicDataType.idl 内で定義されている。

表 4.2 RTC::TimedBoolean

属性		
tm	RTC::Time	時刻情報
data	boolean	ブール値

4.1.3 RTC::TimedString

時刻情報付きの文字列情報を格納するための型。OpenRTM-aist の標準型として BasicDataType.idl 内で定義されている。

表 4.3 RTC::TimedString

属性		
tm	RTC::Time	時刻情報
data	string	文字列情報

4.2 型宣言

本仕様では固有の型宣言の定義は行っていない。

固有の型宣言については、検討課題として継続して検討する。

4.3 コミュニケーション機能用

本仕様では固有のデータ型の定義は行っていない。

5 共通インタフェース定義

以下にコミュニケーション機能共通インタフェースで使用する共通インタフェースの定義を示す。

5.1 データポート

5.1.1 音声データインタフェース

音声入力モジュール、音声フィルタモジュール、音声認識モジュール間で、また対話マネージャモジュールが音声出力モジュールに音声データを受け渡すためのインタフェースである。

音声データは、RTC::TimedOctetSeq 型を用いたバイト列で表現されている。バイト配列の内容については、

2.3.1 音声データを参照のこと。

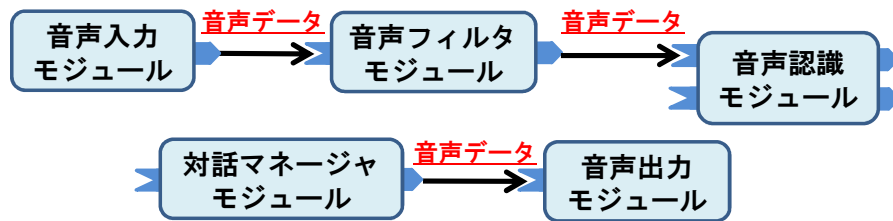


図 5.1 音声データインタフェース

5.1.2 Voice Activity インタフェース

音声認識モジュールに対して、実際の認識処理を実行するかどうかを指定するためのインタフェースである。音声データインタフェースによって伝達された音声データ内に、認識対象となる実際の音声データが含まれているかどうかを指令する。

Voice Activity は RTC::TimedBoolean 型を用いて指定される。



図 5.2 Voice Activity インタフェース

5.1.3 認識ステータスインタフェース

音声認識モジュールから対話マネージャモジュールなどの外部モジュールに、認識状態を伝達するためのインタフェースである。

認識ステータスは、RTC::TimedString 型を用いて以下の内容を入力する。

表 5.1 音声認識状態

出力文字列	内容
LISTEN	入力受付開始
STARTREC	認識処理開始
ENDREC	認識処理終了
REJECTED	入力棄却

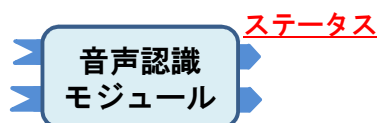


図 5.3 認識ステータスインタフェース

5.1.4 認識結果インタフェース

音声認識モジュールから対話マネージャモジュールに、認識結果を伝達するためのインタフェースである。

認識結果は RTC::TimedString 型を用いた XML 形式の文字列で表現されている。XML スキーマの定義を以下に示す。

音声認識結果は、以下に示す XML 形式によって出力される。XML のスキーマ定義を以下に示す。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="listenText">
    <xs:complexType>
      <xs:all>
        <xs:element minOccurs="1" maxOccurs="unbounded" ref="data"/>
        <xs:element minOccurs="0" maxOccurs="1" ref="abstime"/>
        <xs:element minOccurs="0" maxOccurs="1" ref="utime"/>
        <xs:element minOccurs="0" maxOccurs="1" ref="clipping_ratio"/>
        <xs:element minOccurs="0" maxOccurs="1" ref="snr"/>
        <xs:element minOccurs="0" maxOccurs="1" ref="average_log_power"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="data">
    <xs:complexType>
      <xs:all>
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="word"/>
        <xs:element minOccurs="0" maxOccurs="1" ref="phoneme_model"/>
        <xs:element minOccurs="0" maxOccurs="1" ref="root_rule"/>
      </xs:all>
      <xs:attribute name="rank" use="required" type="xs:integer"/>
      <xs:attribute name="text" use="required" type="xs:string"/>
      <xs:attribute name="score" use="optional" type="xs:decimal"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="word">
    <xs:complexType>
      <xs:attribute name="text" use="required" type="xs:string"/>
      <xs:attribute name="score" use="optional" type="xs:decimal"/>
      <xs:attribute name="time" use="optional" type="xs:decimal"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="abstime" type="xs:decimal"/>
  <xs:element name="utime" type="xs:decimal"/>
  <xs:element name="clipping_ratio" type="xs:decimal"/>
  <xs:element name="snr" type="xs:decimal"/>
  <xs:element name="average_log_power" type="xs:decimal"/>
  <xs:element name="phoneme_model" type="xs:string"/>
  <xs:element name="root_rule" type="xs:string"/>
</xs:schema>
```

表 5.2 XML タグ定義

タグ名		説明
listenText		必須 XML ツリーの開始タグ
data		必須 各認識候補
	rank	必須 アトリビュート: 候補の順位
	text	必須 アトリビュート: 認識文字列[各単語を半角スペース区切り]
	score	任意 アトリビュート: 認識候補の信頼度[0<=x<=1]
word		任意 認識候補を構成する各単語
	text	必須 アトリビュート: 単語文字列
	score	任意 アトリビュート: 各単語の信頼度[0<=x<=1]
	time	任意 アトリビュート: 発話開始からの各単語の開始時刻[秒]
	phonemes	任意 アトリビュート: 単語を構成する音素記号列
abstime		任意 発話開始の UNIX 時刻[秒]
utime		任意 発話区間の時間長[秒]
clipping_ratio		任意 発話区間でのクリッピング発生フレーム比率
snr		任意 非発話区間と発話区間のエネルギー比(任意)
average_log_power		任意 発話区間の平均ログパワー
phoneme_model		任意 認識に用いられた音響モデル名
root_rule		任意 認識に用いられた文法ルール ID

上記スキーマに適合する XML インスタンスの例を以下に示す。

```
<?xml version="1.0" encoding="UTF-8" ?>
<listenText>
  <utime>1.5</utime>
  <data rank="1" score="0.9" text="前に 進んで ください">
    <word score="0.9" time="0" text="前に" />
    <word score="1.0" time="0.5" text="進んで" />
    <word score="0.8" time="1.2" text="ください" />
  </data>
  <data rank="2" score="0.6" text="後ろに 進んで ください">
    <word score="0.1" time="0" text="後ろに" />
    <word score="1.0" time="0.5" text="進んで" />
    <word score="0.8" time="1.2" text="ください" />
  </data>
  <clipping_ratio>0.1</clipping_ratio>
  <snr>0.5</snr>
  <average_log_power>1</average_log_power>
</listenText>
```

5.2 サービスポート

本仕様では固有のサービスポートは定義していない。

6 共通インタフェースを利用したシステム構成例

6.1 対話制御コンポーネント群 OpenHRI

○開発者: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 インタラクションモデリング研究グループ

○詳細 URL:<http://www.openrtm.org/openrtm/ja/project/openhri>

○概要

音声認識・音声合成・対話制御など、ロボットのコミュニケーション機能の実現に必要な各要素を実現するモジュール群。

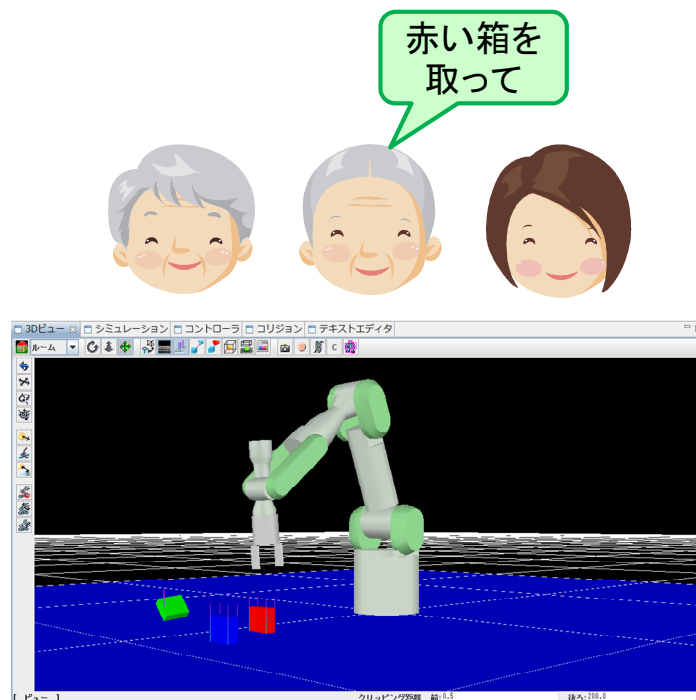


図 6.1 コミュニケーション機能コンポーネント適用例

本モジュール群を利用したシステム構成例を以下に示す。図中の赤字の部分、本仕様書で規定している共通インタフェースを使用している部分である。

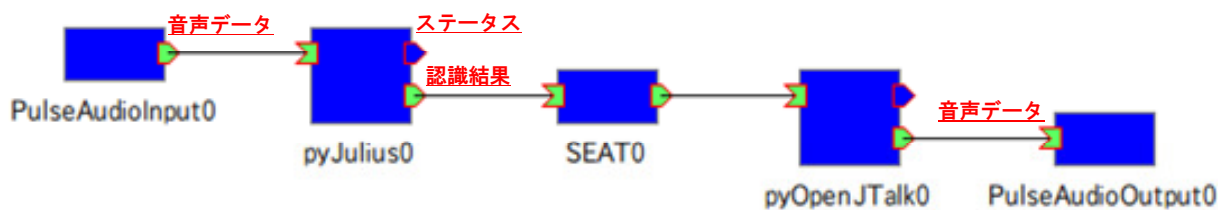


図 6.2 対話制御コンポーネント群 OpenHRI

7 CORBA IDL

コミュニケーション機能共通インタフェースでは、固有の IDL 定義を行っていない。

8 参考文献

なし