

コミュニケーション知能コンポーネント共通規格(第1版)

2010年6月18日

NEDO 次世代ロボット知能化技術開発プロジェクト・コミュニケーション知能 SWG

目次

1はじめに.....	3
2本書を読む上での注意.....	4
2.1 本規格で用いる拡張語彙.....	4
2.1.1 必要なポートに関する拡張語彙.....	4
2.1.2 必要な機能に関する拡張語彙.....	5
3規格の概要.....	6
3.1 コンポーネントの全体構成.....	6
3.2 各コンポーネント共通の規格.....	6
3.2.1 音声のデータ型.....	6
3.2.2 コンフィギュレーションパラメータの適用タイミング.....	7
4各コンポーネント個別の規格.....	8
4.1 音声入力コンポーネント.....	8
4.1.1 入出力ポートとデータ型.....	8
4.1.2 コンフィギュレーションパラメータ.....	8
4.1.3 ドキュメント化の要件.....	8
4.2 音声出力コンポーネント.....	9
4.2.1 入出力ポートとデータ型.....	9
4.2.2 コンフィギュレーションパラメータ.....	9
4.2.3 ドキュメント化の要件.....	9
4.3 音声フィルタコンポーネント.....	10
4.3.1 入出力ポートとデータ型.....	10
4.3.2 コンフィギュレーションパラメータ.....	10
4.3.3 ドキュメント化の要件.....	10
4.4 音声認識コンポーネント.....	11
4.4.1 入出力ポートとデータ型.....	11
4.4.2 コンフィギュレーションパラメータ.....	11
4.4.3 ドキュメント化の要件.....	12
4.4.4 音声認識文法フォーマット.....	13
4.4.5 音声認識結果出力フォーマット.....	15
4.4.6 ステータス出力.....	17
4.5 音声合成コンポーネント.....	18
4.5.1 入出力ポートとデータ型.....	18
4.5.2 コンフィギュレーションパラメータ.....	18
4.5.3 ドキュメント化の要件.....	18
4.5.4 音声合成フォーマット.....	18
4.6 対話マネージャコンポーネント.....	20
4.6.1 入出力ポートとデータ型.....	20
5謝辞.....	21
6履歴.....	21

1 はじめに

近年、ロボットの開発を効率化するため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、インター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、ロボットのヒューマンコミュニケーション機能に関わるコンポーネントの共通規格を定義する。

2 本書を読む上での注意

共通規格化は、規格に合致しないコンポーネントを排除することで、時に開発の制限を作ってしまうこともある。本規格では、そのような制限をなくすために、以下のような方針で規格を定義する。

- 最低限の規格の定義: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規格を定義する。開発の制約となる規格部分は最低限にとどめ、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、本書に書かれた規格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規格から外れるわけではない。

上記の方針による規格の定義を可能にするために、本規格では、以下の語彙を用いる。

2.1 本規格で用いる拡張語彙

本規格では、以下の拡張語彙を用いる。

2.1.1 必要なポートに関する拡張語彙

必要なポートの規格化に関して、以下の語彙を用いる。

one:ひとつのポートを作成すること

one or more:ひとつ以上のポートを作成すること

zero or more:0個以上のポートを作成すること

例えば、以下のような定義が行われた場合、

表 2.1.1.1: ポート定義の例

種類	名前	データ型	説明
入力ポート	任意	TimedOctetSeq	音声データ入力(one)
出力ポート	任意	TimedOctetSeq	音声データ出力 (one or more)

以下に例示するコンポーネントは、この規格を満たすことになる。



図 2.1.1.1: 表 2.1.1.1 の定義に適合するコンポーネント例 1



図 2.1.1.2: 表 2.1.1.1 の定義に適合するコンポーネント例 2

以下に例示するコンポーネントは、この規格を満たさない。



図 2.1.1.3: 表 2.1.1.1 の定義に適合しないコンポーネント例

2.1.2 必要な機能に関する拡張語彙

機能の必要性に関して、以下の 2 つの語彙を用いる。

必須: 実装しなければ規格を満たすことができない。

任意: 実装するかどうかは任意である。

3 規格の概要

3.1 コンポーネントの全体構成

以下に、コミュニケーションに関わるコンポーネントの全体構成を示す。

本書では、コミュニケーションに関わるコンポーネントを以下のカテゴリに分類し、各カテゴリごとに共通規格の定義を行う。

「音声入出力コンポーネント」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力するコンポーネントである。Linux の alsa などの標準インターフェースや、マルチチャンネル A/D, D/A コンバータなどの特殊デバイスに対応したコンポーネントの実装が想定される。

「音声フィルタコンポーネント」は、OpenRTM 形式のデータストリームを入力し、何がし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コンポーネントである。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コンポーネントの実装が想定される。

「音声認識コンポーネント」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コンポーネントのことである。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。

「対話マネージャコンポーネント」は、音声認識コンポーネント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、他のコンポーネントへのコマンドを出力するコンポーネントのことである。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。

「音声合成コンポーネント」は、テキストを入力として、それを音声に変換した後、OpenRTM 形式のデータストリームを出力するコンポーネントである。

3.2 各コンポーネント共通の規格

3.2.1 音声のデータ型

音声のデータ型は、OpenRTM の TimedOctetSeq 型を用いる。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 TimedOctetSeq 型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。

4 各コンポーネント個別の規格

4.1 音声入力コンポーネント

4.1.1 入出力ポートとデータ型

下記のポートを持つものとする。

表 4.1.1.1: ポート定義

種類	名前	データ型	説明
出力ポート	任意	TimedOctetSeq	音声データ出力 (one or more)

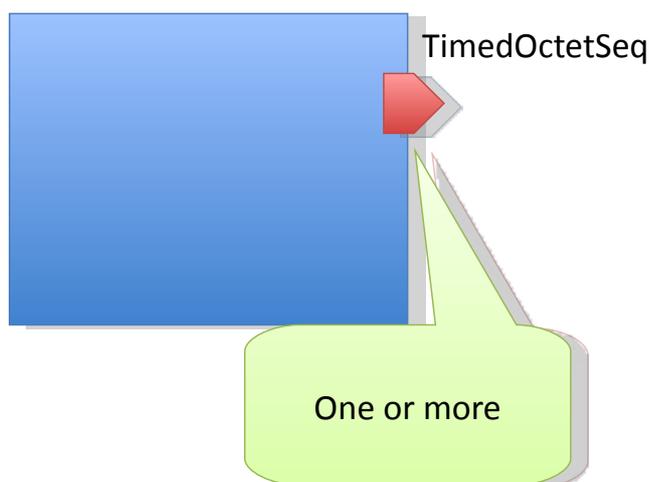


図 4.1.1.1: コンポーネントモデル

4.1.2 コンフィギュレーションパラメータ

サウンドデバイスによっては、様々な種類のサンプルレート、バイトサイズ、チャンネル数で入力可能である。その場合は、以下のコンフィギュレーションパラメータを用いて設定可能としても良い(本機能の実装は任意である)。

表 4.1.2.1: コンフィギュレーションパラメータ一覧

パラメータ名	型	説明
OutputSampleRate	数値型	サンプルレートの設定(任意)
OutputSampleBytes	数値型	バイトサイズの設定(任意)
OutputChannelNumbers	数値型	チャンネル数の設定(任意)

4.1.3 ドキュメント化の要件

各コンポーネントは、以下の内容をドキュメントに含める必要がある。

- 各ポートが出力するデータのサンプルレート(必須)
- 各ポートが出力するデータのバイトサイズ(必須)
- 各ポートが出力するデータのチャンネル数(必須)

4.2 音声出力コンポーネント

4.2.1 入出力ポートとデータ型

下記のポートを持つものとする。

表 4.2.1.1: ポート定義

種類	名前	データ型	説明
入力ポート	任意	TimedOctetSeq	音声データ入力 (one or more)

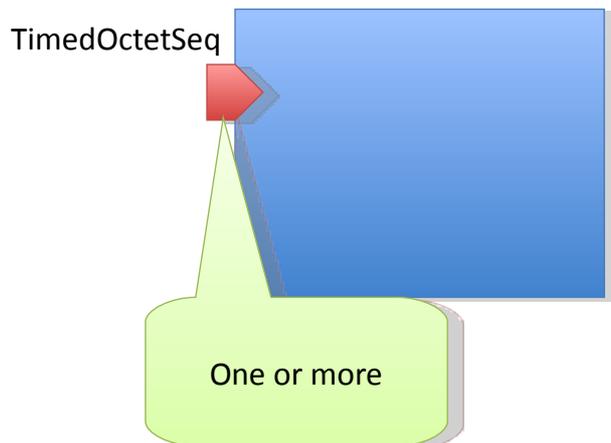


図 4.2.1.1: コンポーネントモデル

4.2.2 コンフィギュレーションパラメータ

サウンドデバイスによっては、様々な種類のサンプルレート、バイトサイズ、チャンネル数で出力可能である。その場合は、以下のコンフィグレーションパラメータを用いて設定可能としても良い(本機能の実装は任意である)。

表 4.2.2.1: コンフィグレーションパラメータ一覧

パラメータ名	型	説明
InputSampleRate	数値型	サンプルレートの設定(任意)
InputSampleBytes	数値型	バイトサイズの設定(任意)
InputChannelNumbers	数値型	チャンネル数の設定(任意)

4.2.3 ドキュメント化の要件

各コンポーネントは、以下の内容をドキュメントに含める必要がある。

- 各ポートが入力可能なデータのサンプルレート(必須)
- 各ポートが入力可能なデータのバイトサイズ(必須)
- 各ポートが入力可能なデータのチャンネル数(必須)

4.3 音声フィルタコンポーネント

4.3.1 入出力ポートとデータ型

下記のポートを持つものとする。

表 4.3.1.1: ポート定義

種類	名前	データ型	説明
入力ポート	任意	TimedOctetSeq	音声データ入力 (one or more)
出力ポート	任意	TimedOctetSeq	音声データ出力 (one or more)

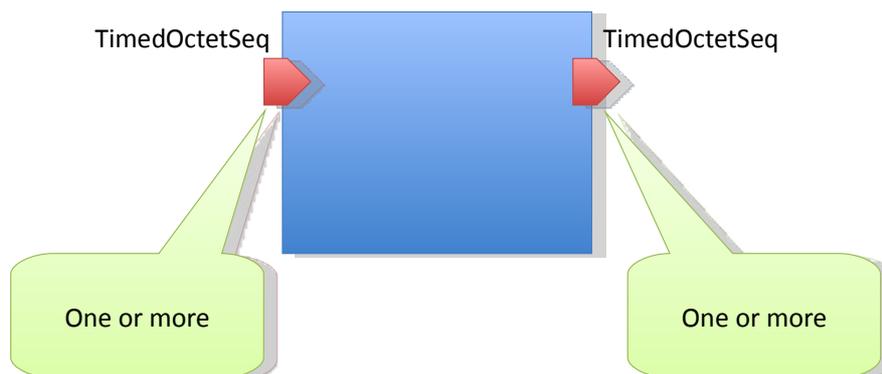


図 4.3.1.1: コンポーネントモデル

4.3.2 コンフィギュレーションパラメータ

アルゴリズムの実装によっては、様々な種類のサンプルレート、バイトサイズ、チャンネル数で入出力可能にすることもできる場合がある。その場合は、以下のコンフィギュレーションパラメータを用いて設定可能としても良い(本機能の実装は任意である)。

表 4.3.2.1: コンフィギュレーションパラメータ一覧

パラメータ名	型	説明
InputSampleRate	数値型	サンプルレートの設定(任意)
InputSampleBytes	数値型	バイトサイズの設定(任意)
InputChannelNumbers	数値型	チャンネル数の設定(任意)
OutputSampleRate	数値型	サンプルレートの設定(任意)
OutputSampleBytes	数値型	バイトサイズの設定(任意)
OutputChannelNumbers	数値型	チャンネル数の設定(任意)

4.3.3 ドキュメント化の要件

各コンポーネントは、以下の内容をドキュメントに含める必要がある。

- 各ポートが入力可能なデータのサンプルレート(必須)
- 各ポートが入力可能なデータのバイトサイズ(必須)
- 各ポートが入力可能なデータのチャンネル数(必須)
- 各ポートが出力するデータのサンプルレート(必須)
- 各ポートが出力するデータのバイトサイズ(必須)
- 各ポートが出力するデータのチャンネル数(必須)

4.4 音声認識コンポーネント

4.4.1 入出力ポートとデータ型

下記のポートを持つものとする。

表 4.4.1.1: ポート定義

種類	名前	データ型	説明
入力ポート	任意	TimedOctetSeq	音声データ入力 (one)
入力ポート	任意	TimedBoolearn	Voice Activity 入力 (one, 任意)
出力ポート	任意	TimedString	認識結果出力 (one)
出力ポート	任意	TimedString	ステータス出力 (one)

4.4.2 コンフィギュレーションパラメータ

アルゴリズムの実装によっては、様々な機能をもたせることができる場合がある。その場合は、以下のコンフィギュレーションパラメータを用いて設定可能としても良い(本機能の実装は任意である)。

表 4.4.2.1: コンフィグレーションパラメーター一覧

パラメータ名	型	説明
InputSampleRate	数値型	入力サンプルレートの設定 (任意)
InputSampleBytes	数値型	入力バイトサイズの設定 (任意)
InputChannelNumbers	数値型	入力チャンネル数の設定 (任意)
PhonemeModel	文字列型 ("male", "female" など)	音響モデルの設定 (任意)
VoiceActivityDetection	文字列型 ("internal", "external")	Voice Activity の検出を内部で行うか外部入力とするかの設定 (任意)
Dictation	文字列型 ("disable", "enable")	ディクテーション機能の有効化 (任意)
Grammar	文字列型	音声認識文法の内容 (任意)
RootRule	文字列型	認識を開始する文法ルール ID (任意) ※本パラメータに限りコンポーネントが Active 状態でも設定できる。
OutputABSTime	文字列型 ("disable", "enable")	絶対時間出力の有効化 (任意)
OutputUTime	文字列型 ("disable", "enable")	発話区間長出力の有効化 (任意)
OutputAverageLogPower	文字列型 ("disable", "enable")	平均ログパワー出力の有効化 (任意)
OutputSNR	文字列型 ("disable", "enable")	非発話区間と発話区間のパワー比出力の有効化 (任意)
OutputClippingRatio	文字列型 ("disable", "enable")	発話区間内でのクリッピング発生比率出力の有効化 (任意)
OutputPhonemeModel	文字列型 ("disable", "enable")	認識に用いた音響モデル名出力の有効化 (任意)
OutputRootRule	文字列型 ("disable", "enable")	認識に用いた文法ルール ID 出力の有効化 (任意)
OutputScore	文字列型 ("disable", "enable")	スコア出力の有効化 (任意)
OutputWords	文字列型 ("disable", "enable")	単語出力の有効化 (任意)
OutputWordScores	文字列型 ("disable", "enable")	単語スコア出力の有効化 (任意)
OutputWordTimes	文字列型 ("disable", "enable")	単語時刻出力の有効化 (任意)

4.4.3ドキュメント化の要件

各コンポーネントは、以下の内容をドキュメントに含める必要がある。

- ポートが入力可能なデータのサンプルレート (必須)
- ポートが入力可能なデータのバイトサイズ (必須)
- ポートが入力可能なデータのチャンネル数 (必須)

4.4.4 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。

コンポーネントは、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/> S) 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|こんにちは}}</phoneme>
  </lexeme>
  <lexeme>
    <grapheme>こんにちは</grapheme>
    <phoneme>{{KANA|こんにちは}}</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なると実用的であろう。音声認識文法は、コンフィギュレーションパラメータによってその内容を文字列型としてコンポーネントに登録するか、起動時のコマンドラインによってファイルを指定することでコンポーネントに登録する。

4.4.5 音声認識結果出力フォーマット

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>

```

上記スキーマによって生成される XML 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>

```

XML の各タグについて説明する。

表 4.4.5.1:コンフィグレーションパラメーター一覧

タグ名	説明
listenText	XML ツリーの開始タグ (必須)
data	各認識候補 (必須) rank アトリビュート: 候補の順位 (必須) text 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] (必須) score アトリビュート: 認識候補の信頼度 [0<=x<=1] (任意)
word	認識候補を構成する各単語 (任意) text アトリビュート: 単語文字列 (必須) score アトリビュート: 各単語の信頼度 [0<=x<=1] (任意) time アトリビュート: 発話開始からの各単語の開始時刻 [秒] (任意) phonemes アトリビュート: 単語を構成する音素記号列 (任意)
abstime	発話開始の UNIX 時刻 [秒] (任意)
utime	発話区間の時間長 [秒] (任意)
clipping_ratio	発話区間でのクリッピング発生フレーム比率 (任意)
snr	非発話区間と発話区間のエネルギー比 (任意)
average_log_power	発話区間の平均ログパワー (任意)

phoneme_model	認識に用いられた音響モデル名(任意)
root_rule	認識に用いられた文法ルール ID(任意)

4.4.6 ステータス出力

ステータスポートは以下のテキストを出力する。

“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

4.5 音声合成コンポーネント

4.5.1 入出力ポートとデータ型

下記のポートを持つものとする。

表 4.5.1.1:ポート定義

種類	名前	データ型	説明
入力ポート	任意	TimedString	音声合成フォーマットによるテキスト入力 (one)
出力ポート	任意	TimedOctetSeq	音声データ出力 (one)

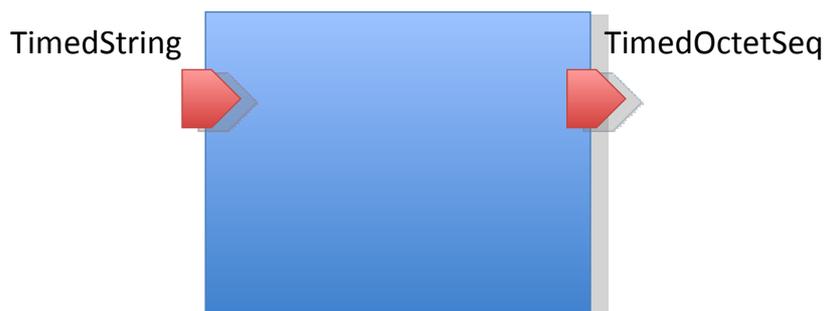


図 4.5.1.1: コンポーネントモデル

4.5.2 コンフィギュレーションパラメータ

アルゴリズムの実装によっては、様々な種類のサンプルレート、バイトサイズ、チャンネル数で出力可能にすることもできる場合がある。その場合は、以下のコンフィギュレーションパラメータを用いて設定可能としても良い(本機能の実装は任意である)。

表 4.5.2.1:コンフィギュレーションパラメータ一覧

パラメータ名	型	説明
OutputSampleRate	数値型	サンプルレートの設定(任意)
OutputSampleBytes	数値型	バイトサイズの設定(任意)
OutputChannelNumbers	数値型	チャンネル数の設定(任意)

4.5.3 ドキュメント化の要件

各コンポーネントは、以下の内容をドキュメントに含める必要がある。

- ポートが出力するデータのサンプルレート(必須)
- ポートが出力するデータのバイトサイズ(必須)
- ポートが出力するデータのチャンネル数(必須)

4.5.4 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。

以下にその例を示す。

```
<?xml version="1.0" encoding="UTF-8"?>
<speech 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>
```

4.6 対話マネージャコンポーネント

4.6.1 入出力ポートとデータ型

下記のポートを持つものとする。

表 4.6.1.1: ポート定義

種類	名前	データ型	説明
入力ポート	任意	TimedString	音声認識結果入力 (one or more)
入力ポート	任意	任意	その他のセンサ入力 (zero or more)
出力ポート	任意	TimedString	音声合成テキスト出力 (one or more)
出力ポート	任意	任意	コマンド出力 (zero or more)

5 謝辞

本規格の作成にあたっては、ATRの萩田紀博氏、宮下敬宏氏、石井カルロス寿憲氏、佐藤幹氏、NECの高野陽介氏、山下信行氏、九州工業大学の大橋健先生、嶋田和孝氏から多くの有用な意見をいただいた。また、認識結果のXML出力に関してはATR、認識文法規格に関してはNECなど、規格の多くの重要な部分において、NEDO次世代ロボット知能化技術開発プロジェクトにおいて開発された、ATR、NEC、九州工業大学の各コンポーネント規格を引用させていただいた。本共通規格の策定に協力いただいた関係諸氏に深く感謝する。

6 履歴

第1版（2010-06-18） 編集者：産業技術総合研究所・松坂要佐