

ChoreonoidとOpenHRIについて

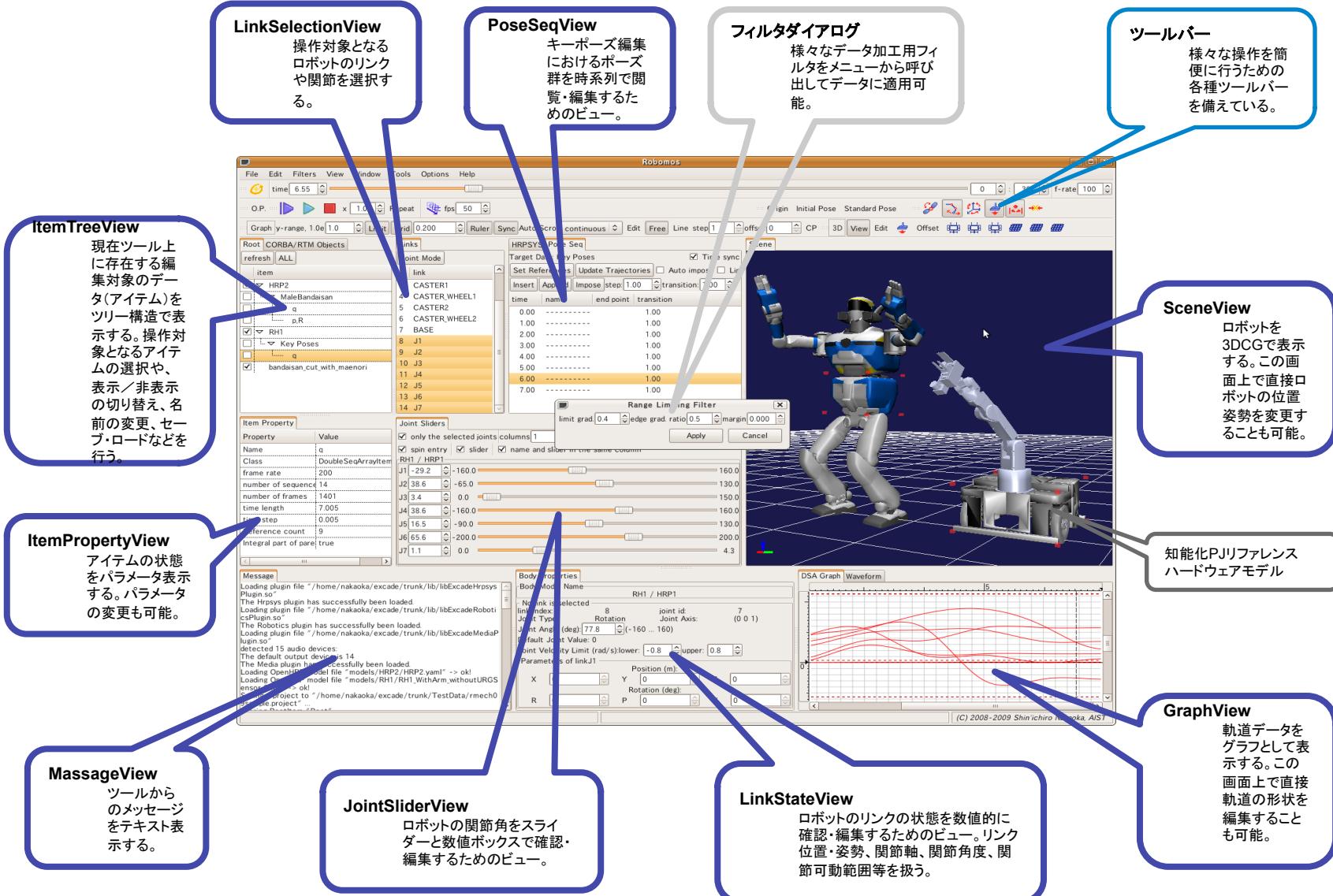
2011/07/06 13:00-14:30

産総研 原 功

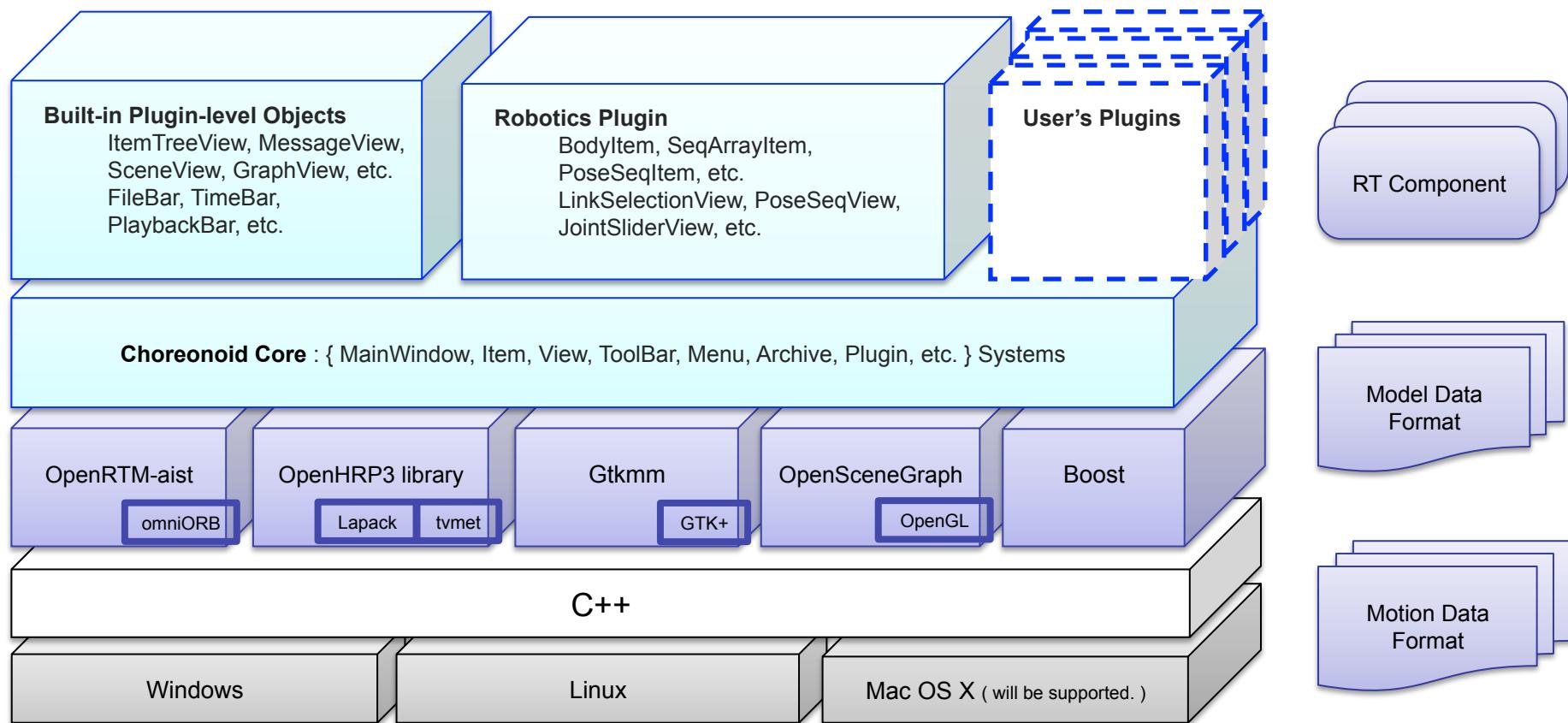
講義の内容

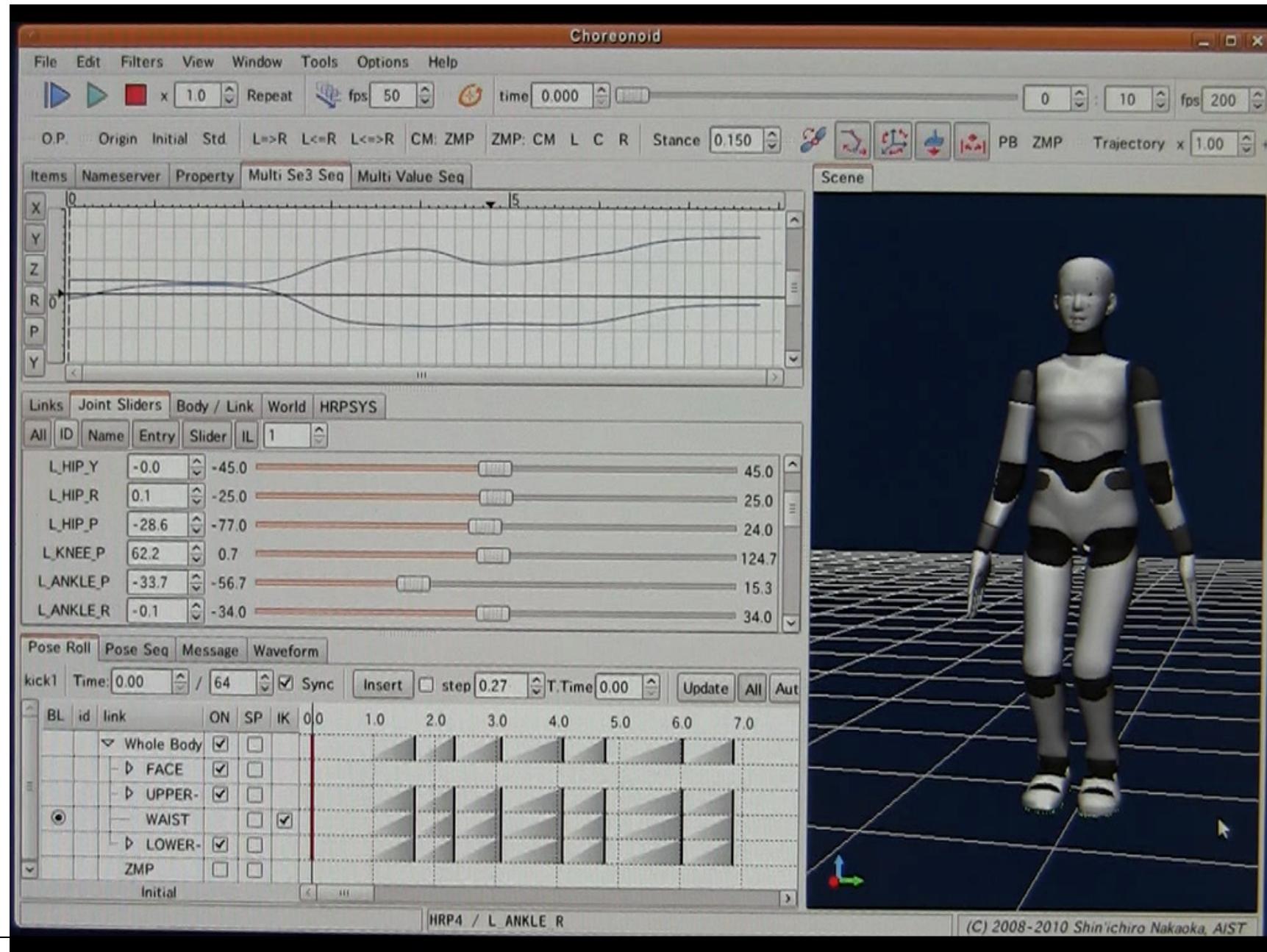
- Choreonoidの紹介
 - Choreonoid の概要
 - Choreonoidの活用事例
- OpenHRIの紹介
 - OpenHRIで提供される機能
 - OpenHRIを使ったコミュニケーションロボットの事例

- 知能化PJにおいて開発したロボット知能ソフトウェアプラットフォームの1つのツール
 - 多関節型ロボットの動作パターンを作成するためのGUIツール
 - 直感的なGUI
 - キーフレームベースの姿勢設定と動作補完
 - 姿勢設定時に動力学シミュレーションを同時実行



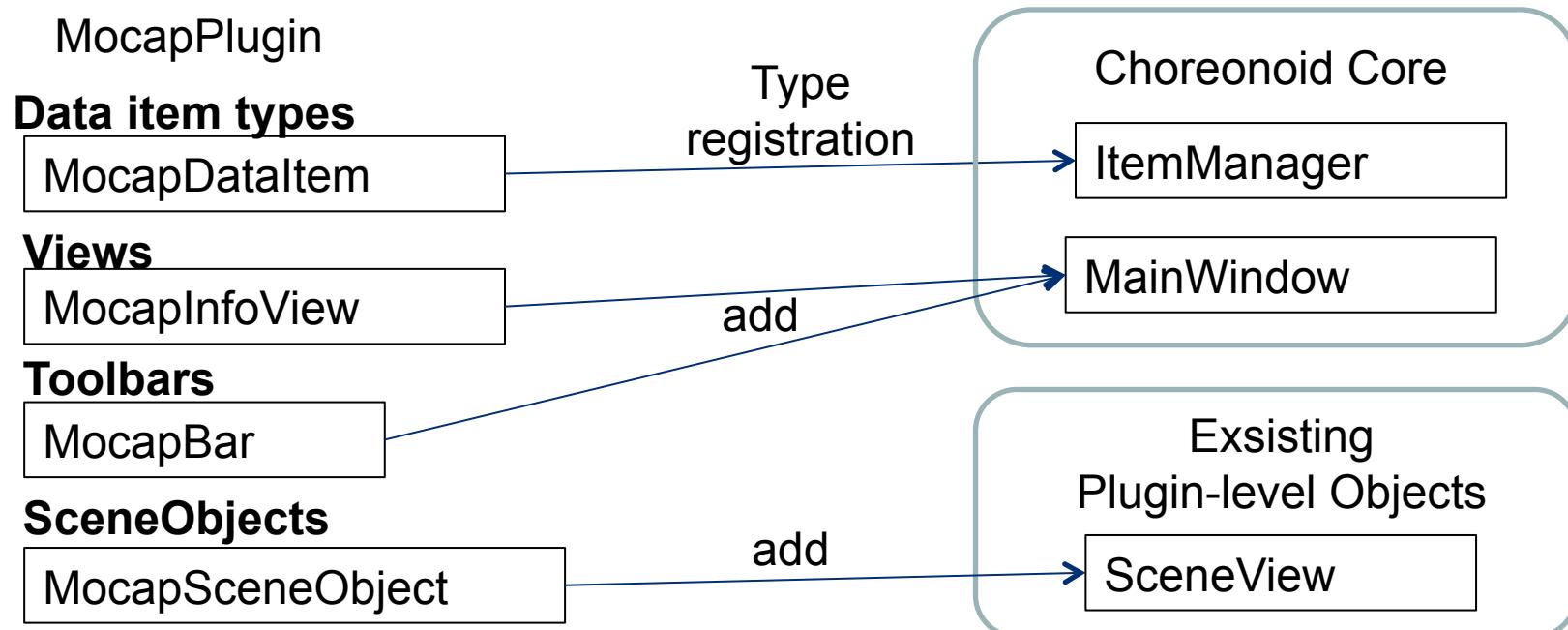
- ・ソフトウェア構成





- ・ プラグインシステム

- 独自の編集対象データ形式やGUIインターフェースを自由に追加し、既存のオブジェクトと連携させることが可能。上図ではモーションキャプチャデータを扱うプラグインを想定。



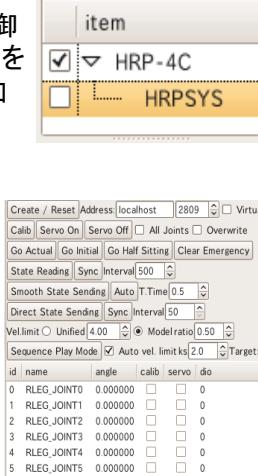
• HRP-4C 顔表情パターン編集システム

ツール上で作成した表情をリアルタイムで実ロボットに同期させるための”Hrpsysプラグイン”と、表情をセリフの音声にマッチさせるための”Audioプラグイン”を新たに作成し、効率的な表情パターン作成を可能とした。

HRPSYSプラグイン

Hrpsystem:
HRP-4 体内PCで動作する制御システム “HRPSYS”との通信を管理。BodyItem の更新を閲覧して関節角等を実ロボットへ転送する。

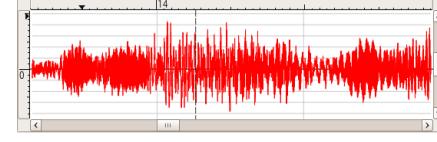
HrpsysView:
HrpsysItem の状態表示、操作用GUIインターフェースを提供。



AudioPlugin

Audiolitem:
音声データの保持と再生を行う。WAVファイルを読み込み可能。再生倍率変更にも対応。

WaveformView:
音声データの波形を表示する。音声同期を視覚的に支援する。



キーポーズ列作成

姿勢編集のための各種機能を用いて、キーポーズ列を作成する。

time	name	end point	transition
0.00	Normal Face	1.00	
2.50	surprise	0.30	
4.50	mabataki	0.50	
5.00	surprise	0.30	
6.50	mabataki2	0.20	
6.90	surprise	0.30	
7.30	mabataki2	0.20	
7.70	surprise	0.30	
10.00	Normal Face	1.00	



File Edit Filters View Window Tools Options Help

x 1.0 Repeat fps 50 time 0.000 0 : 220

O.P. Body Motion Auto Balancer Setup C P Origin Initial Std. L=>R L<=R L<=>R CM: C ZMP ZMP: CM L C R Stance 0.1369 SE3

Scene X World Scenario Property Media X Items Links HRPSYS Joint Sliders Body / Link

Message Nameserver Waveform Multi Value Seq Multi Se3 Seq Pose Seq Pose Roll X

Menu T: 0.000 / 210 Sync Insert TT: 0.000 Update All Auto T: 0.000 TT: 0.000 Delete Grid: 1

BL	id	link	ON	SP	IK		0.0	1.0	2.0	3.0
		Whole Body	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
		FACE	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
		UPPER-BODY	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
		WAIST	<input type="checkbox"/>	<input checked="" type="checkbox"/>						
		LOWER-BODY	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
		ZMP	<input type="checkbox"/>	<input type="checkbox"/>						

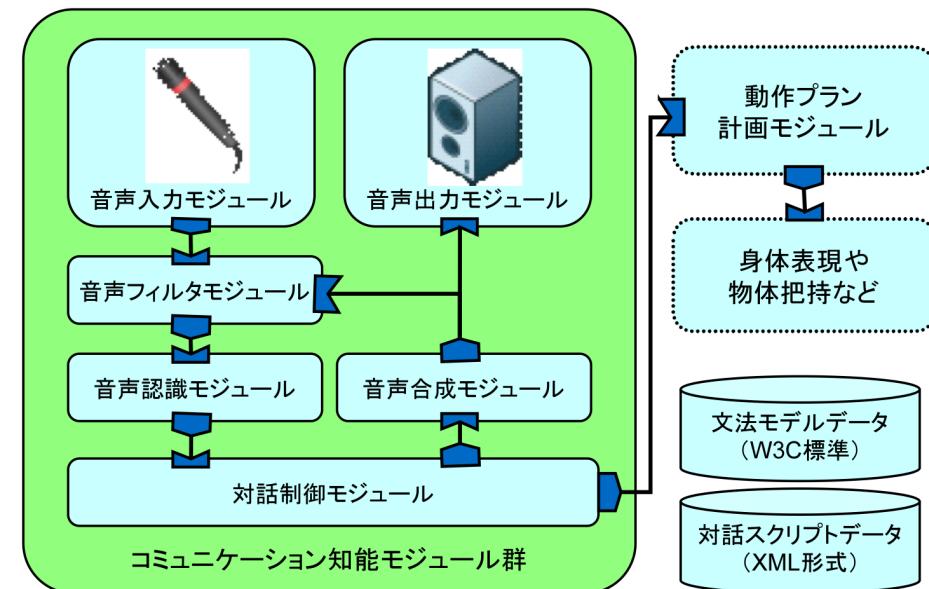
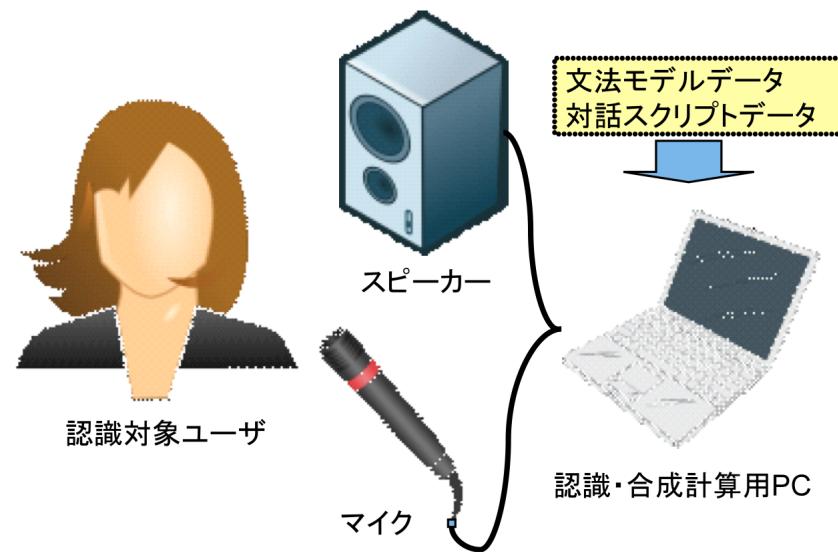
Deattatoki11 :

講義の内容

- **Choreonoidの紹介**
 - Choreonoid の概要
 - Choreonoidの活用事例
- **OpenHRIの紹介**
 - OpenHRIで提供される機能
 - OpenHRIを使ったコミュニケーションロボットの事例

- OpenHRIは、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コンポーネント群
- フリーで利用できる各オープンソースソフトウェアを使い易いコンポーネントとしてまとめたもの
- 知能化PJにおいて開発され、EPL-1.0にて公開
<http://openhri.net>

- ・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
(音声入出力・音声認識・音声合成・対話制御など)

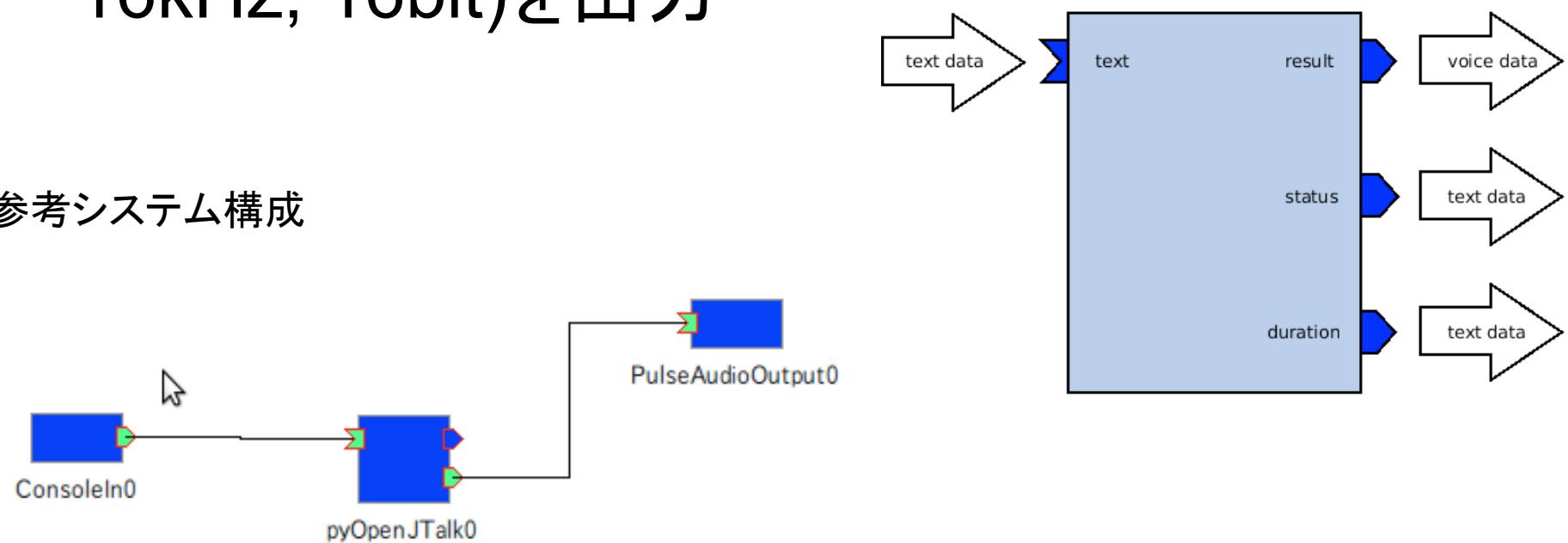


※IFは全て各社共通形式(どのモジュールでも差し替え可能です)

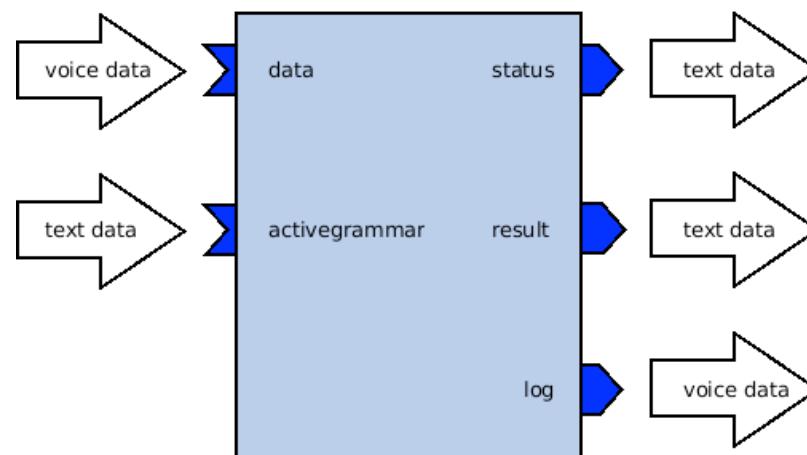
- バイナリパッケージ動作環境
 - WindowsXP以上またはUbuntu Linux 10.04においてOpenRTM-aist-1.0(Python)が動作し、RTSエディタ等が動作すること
 - Windows用インストーラとUbuntu Linux 10.04のSynapticパッケージ・マネージャに対応
 - 音声入力装置(マイクロホン等)および音声出力装置(スピーカ等)が必要

- ・ 音声合成コンポーネントは、フリーで利用できる音声合成エンジン(OpenJTalk、Festival)、を利用
- ・ テキストを入力とし、音声データ(Mono, 16kHz, 16bit)を出力

参考システム構成



- ・ オープンソースの高性能な汎用大語彙連続音声認識エンジンJuliusを使った音声認識機能
- ・ 受け取った音声データ(16bits, 16KHz)を音声認識して認識テキストに変換する
- ・ 音声認識は、予め用意された音声認識文法ファイルに従って実行される

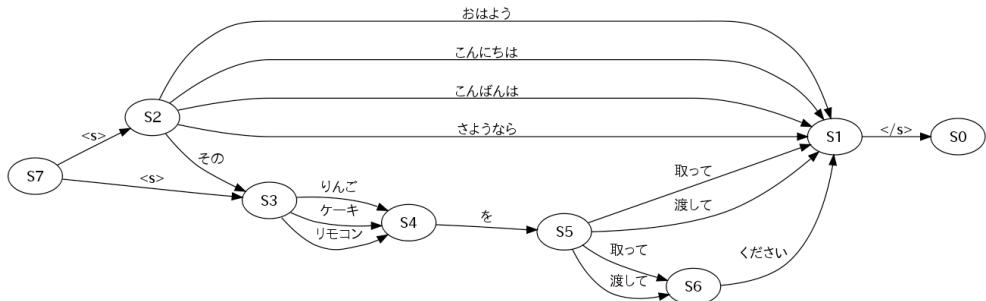


- 音声認識文法
 - 音声認識文法は、音声認識器に与える認識可能な文法(単語の構造)を定義したもの
 - W3C-Speech Recognition Grammar Specificationに準拠した記述を使用
 - 検証ツール: validatesrgs
 - 視覚化ツール: srgstojulius

- 音声認識文法の作成
 - OpenHRIの音声認識RTCでは、W3C-SRGS形式の音声認識文法を使用
 - W3C-SRGSのタグ
 - **lexicon**: W3C-PLS辞書(次のセクション)のURIを定義します。任意。
 - **rule**: IDによって区別された各文法を定義します。IDは音声認識文法の相互参照や、Julius音声認識コンポーネントによって認識されるアクティブな文法を切り換えるのに利用します。
 - **item**: 認識される単語や文を定義します。repeatプロパティで繰り替えされる回数を指定できます。
 - **one-of**: 子項目で定義される文法がすべて許容できることを示します。
 - **ruleref**: uriで指定される文法を参照します。

- 音声認識文法の例
 - 簡単な挨拶
 - 手渡し依頼

視覚化ツールによる出力



```

<?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="main">
<lexicon uri="sample-lex-jp.xml"/>
<rule id="main">
  <one-of>
    <item><ruleref uri="#greet" /></item>
    <item><ruleref uri="#command" /></item>
  </one-of>
</rule>
<rule id="greet">
  <one-of>
    <item>おはよう</item>
    <item>こんにちは</item>
    <item>こんばんは</item>
    <item>さようなら</item>
  </one-of>
</rule>
<rule id="command">
  <item repeat="0-1">その</item>
  <one-of>
    <item>りんご</item>
    <item>ケーキ</item>
    <item>リモコン</item>
  </one-of>
  <item>を</item>
  <one-of>
    <item>取って</item>
    <item>渡して</item>
  </one-of>
  <item repeat="0-1">ください</item>
</rule>
</grammar>
  
```

- SEAT(Speech Event Action Transfer)とは?
 - ロボットとの対話を実現するための小さく軽量な対話エンジン
 - ロボットの対話動作は、音声認識結果(条件)とシステムの動作(アクション)のペアを基本として対話動作を記述
 - 状態遷移モデルに基づく対話管理
 - XMLベースのSEATMLにより対話スクリプトを記述

SEATMLとは？

SEATMLは、シンプルな対話エンジンSEATの動作を定義するためのXMLファイル。状態ごとの<条件ーアクション>のルールを記述し、条件に適合した場合の動作を記述したもの。

- アダプタの定義

- SEATには、名前と通信方法(RTMとソケット)を対応付けるアダプタ機構を持っています。アダプタ機構は、通信方法の差異を隠蔽化することで、システムのハードウェア構成の変化に適応し、対話口语の再利用性を向上させます。
- General : アダプタ定義部を示します。
- Agent : 名前と通信方法の対応を示します。“type”属性は”rtcin”、“rtcout”、“socket”を取ることができます。タイプが”rtcin”か”rtcout”と定義されたとき、“datatype”属性を定義できます(データ型に関しては、RTMの仕様を参照してください)。タイプが”socket”と定義されたとき、“host”、“port”属性を定義できます。

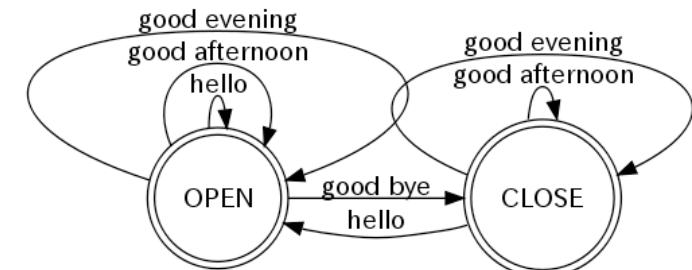
- スクリプト定義

- State: 状態遷移モデルで状態を示します。
- Rule : キーワードとコマンドの組を定義します。
- Key : キーワードを示します。
- Command : キーワードと入力が一致したとき実行されるコマンドを示します。
- Statetransition : 状態遷移を示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<seatml>
  <general name="sample">
    <agent name="speechin" type="rtcin" datatype="TimedString" />
    <agent name="speechout" type="rtcout" datatype="TimedString" />
  </general>

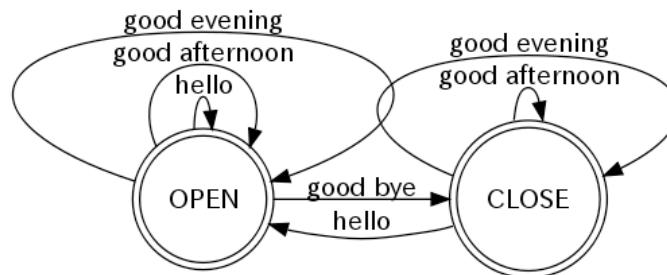
  <state name="OPEN">
    <rule>
      <key>hello</key> <command host="speechout">Hello.</command>
    </rule>
    <rule>
      <key>good afternoon</key> <command host="speechout">Good afternoon.</command>
    </rule>
    <rule>
      <key>good evening</key> <command host="speechout">Good evening.</command>
    </rule>
    <rule>
      <key>good bye</key> <command host="speechout">Good bye.</command> <statetransition>CLOSE</statetransition>
    </rule>
  </state>

  <state name="CLOSE">
    <rule>
      <key>hello</key> <command host="speechout">Hello there.</command> <statetransition>OPEN</statetransition>
    </rule>
    <rule>
      <key>good afternoon</key> <command host="speechout">I'm not available.</command>
    </rule>
    <rule>
      <key>good evening</key> <command host="speechout">I'm not available.</command> </rule>
    </state>
  </seatml>
```

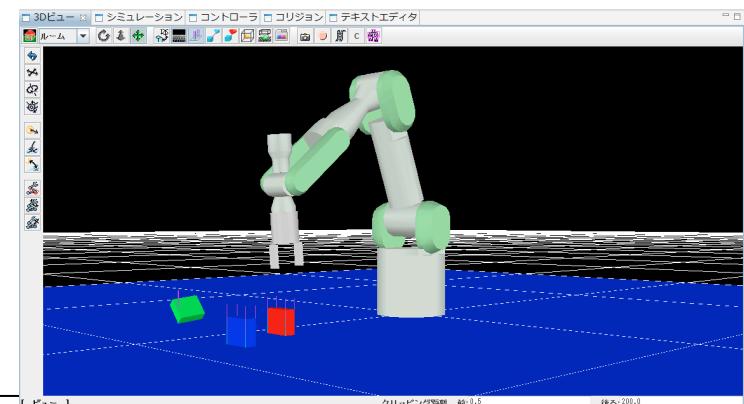


OpenHRIでは、SEATMLを記述するために下記のツールを提供

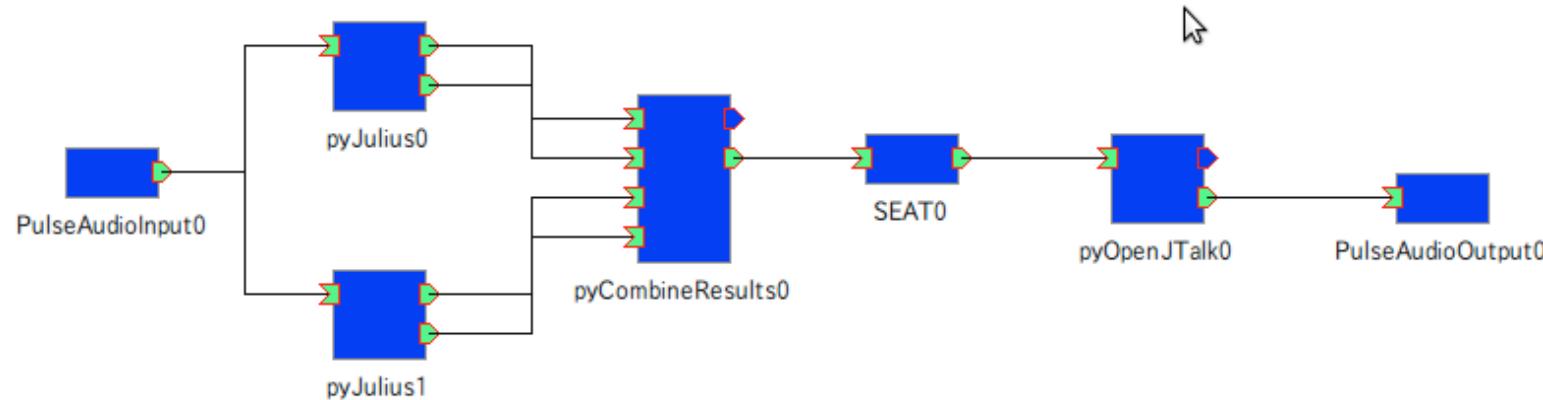
- 検証ツール: validateseatml
 - SEATMLの文法チェックコマンド
- 視覚化ツール: seatmltographviz
 - SEATMLスクリプトをグラフ表示させて構造をチェックするツール



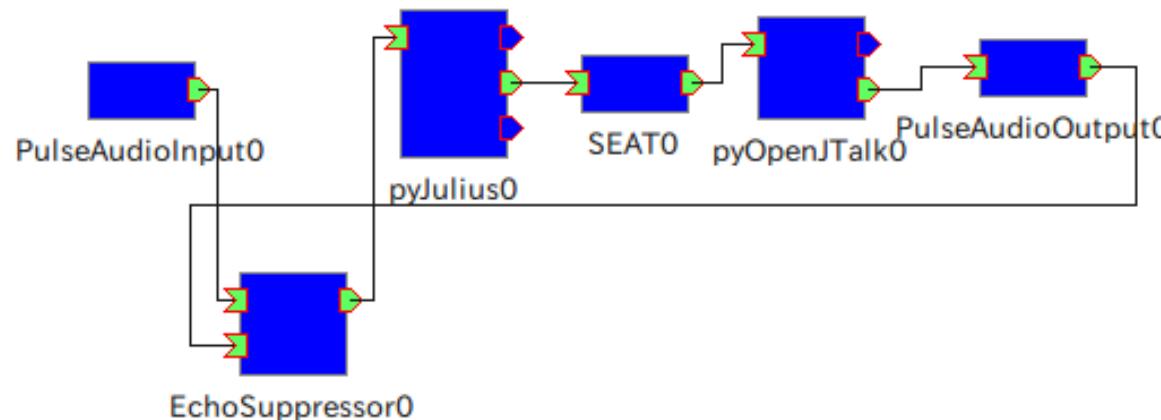
- 音響フィルタコンポーネント
 - **EchoSuppressor**: 音声出力へデータを送信しているとき、音声入力から収音したデータを減衰させる
 - **EchoCanceler**: 音声出力信号からエコー成分を推定し、音声入力で収音したデータから推定したエコー成分を削除
 - **NoiseReduction**: 音声信号から指定した周波数の周辺帯域を強調する
 - **SamplingRateConverter**: 音声入力のサンプリング周波数を変換する
- OpenHRP3を使ったワールドシミュレータ
 - ロボットとのインタラクション機能をテストすることができるワールドシミュレータ



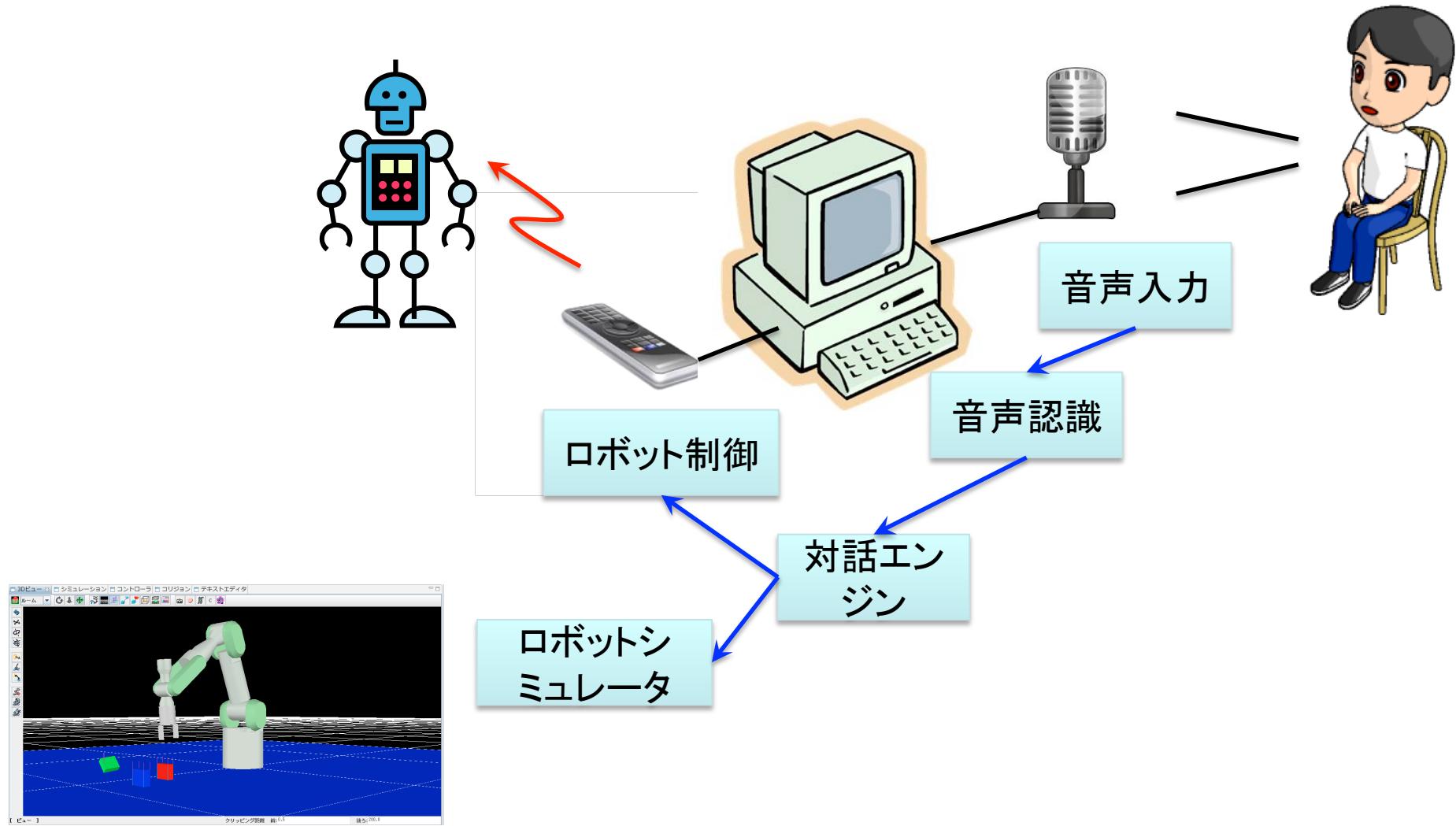
- マルチリンガル対話システム



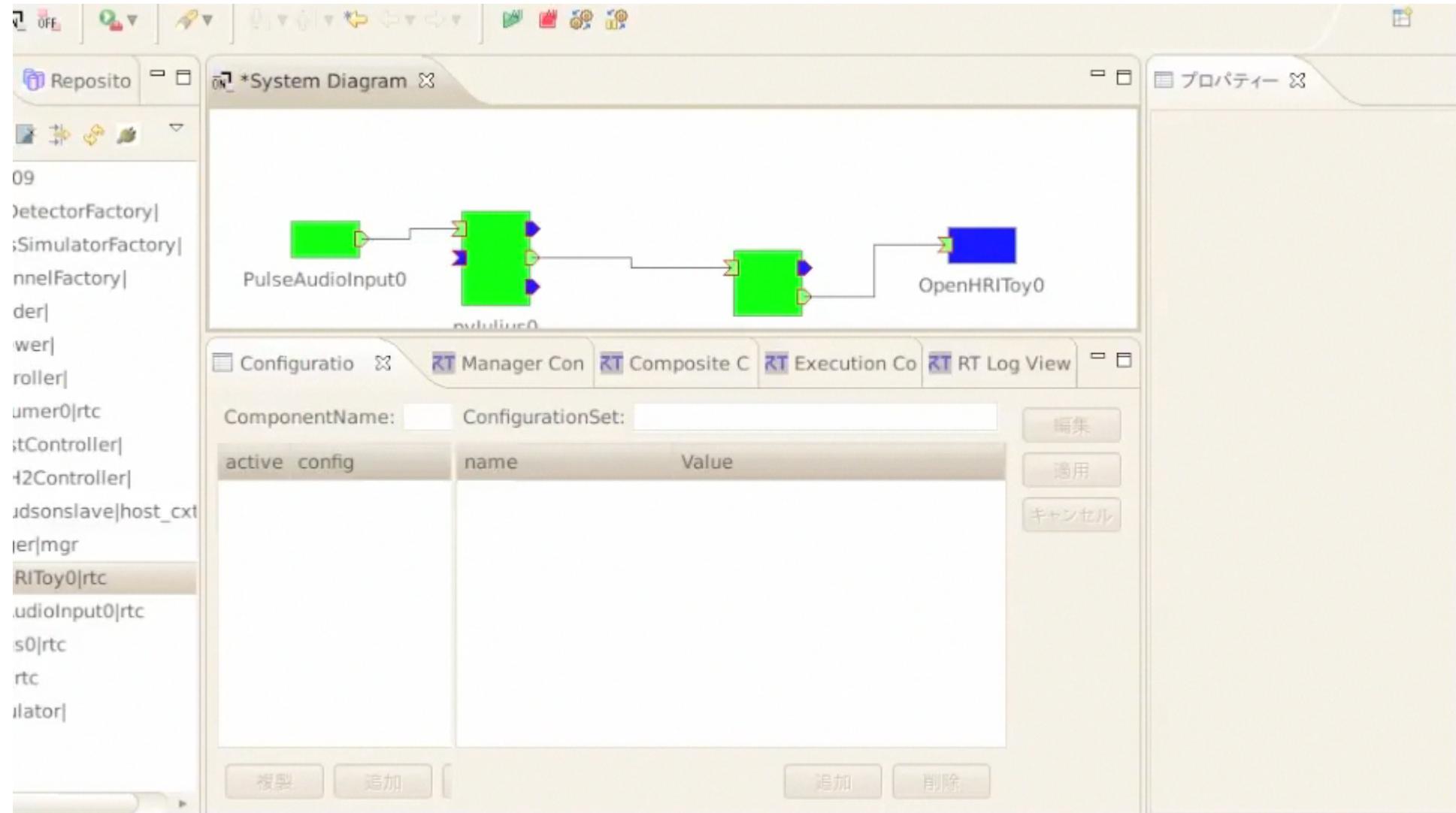
- 音響エコー除去機能を有する対話システム構成



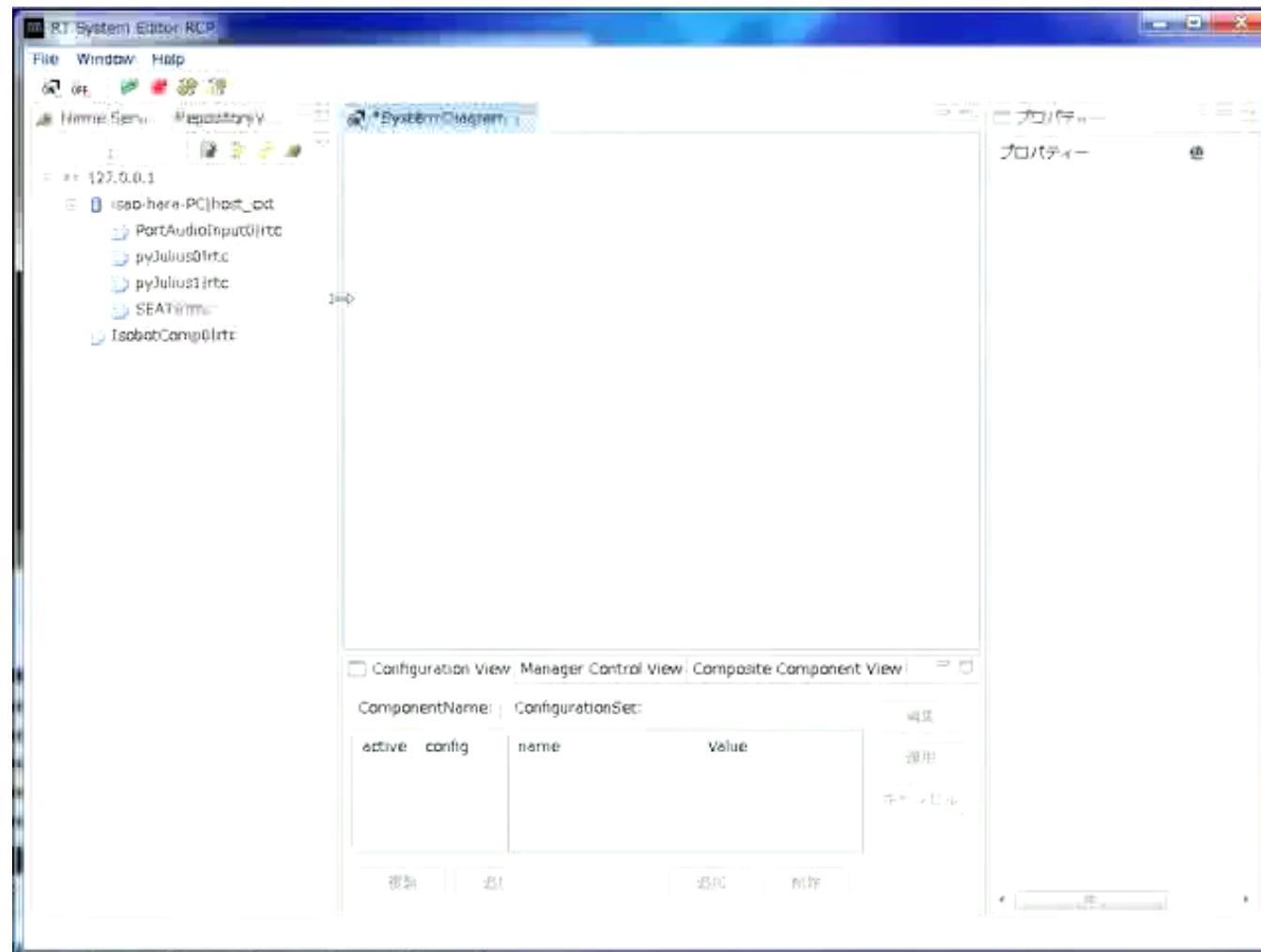
- ・ロボットに音声インターフェースをつける



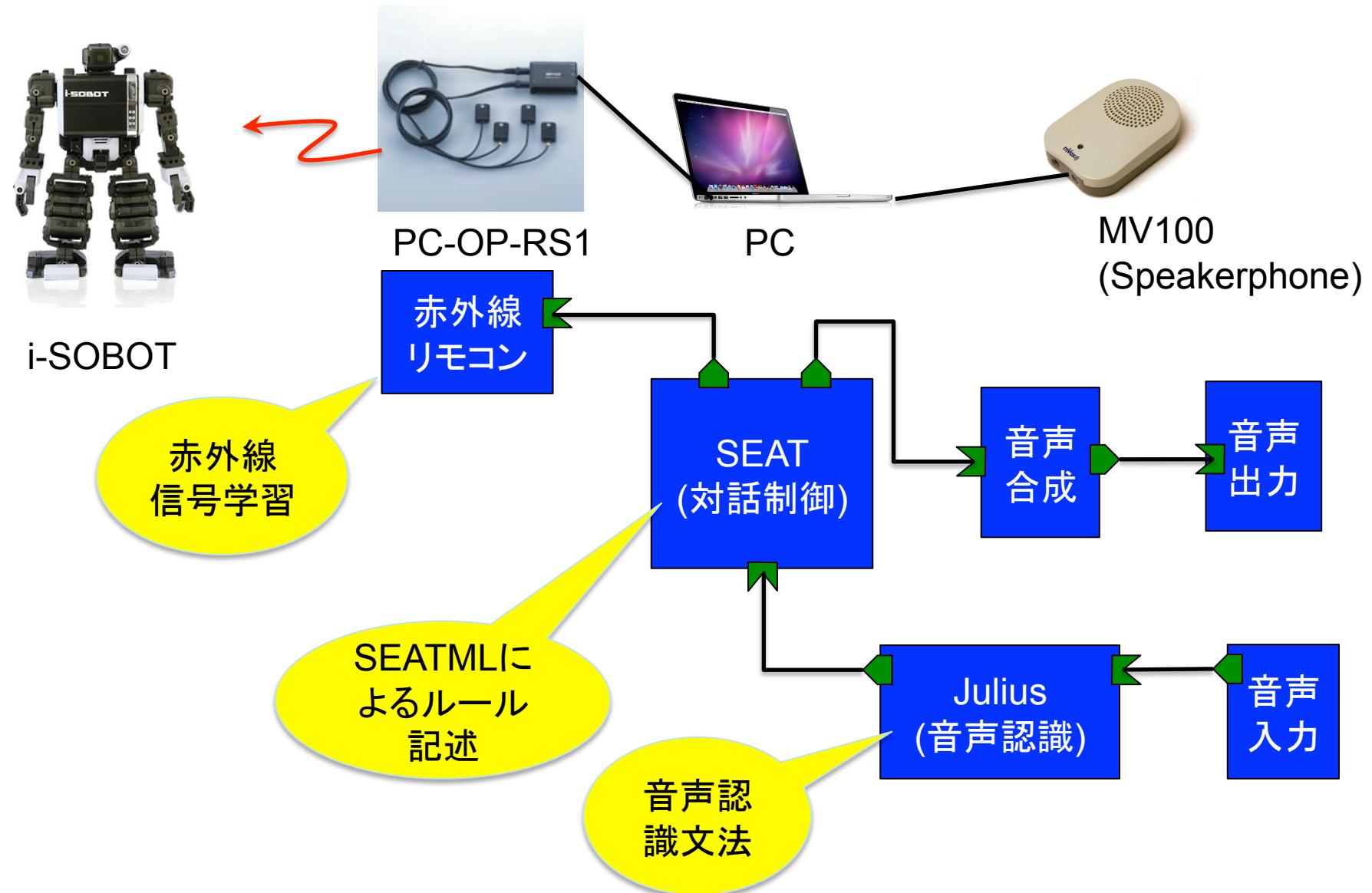
- ・ロボットシミュレータを使った音声インターフェース



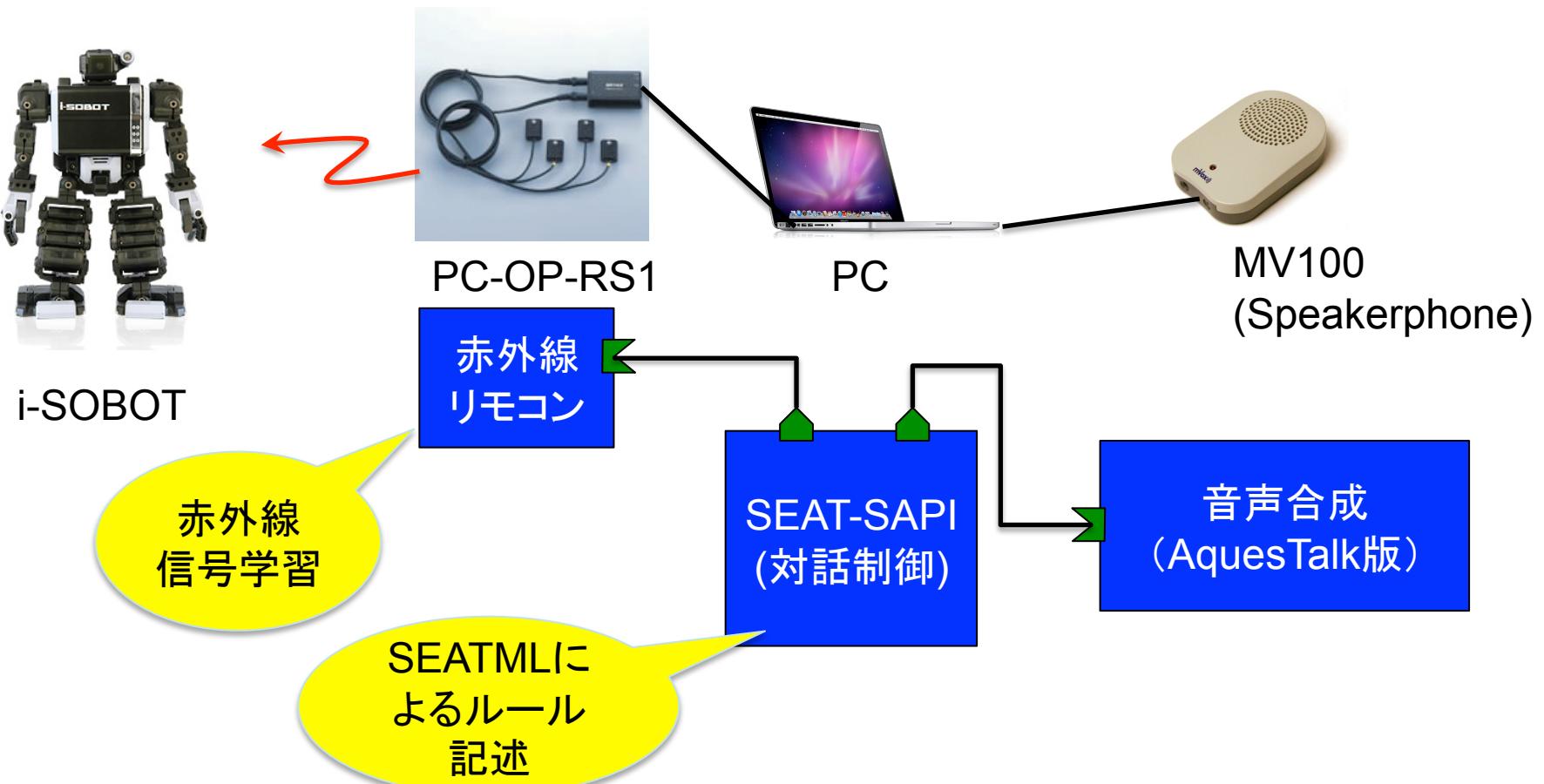
- 音声によるi-SOBOTの操作



本来は、、、、



- 2009年11月国際ロボット展にて実施
- OpenHRIの旧版を使用。Windows7の音声認識を利用。
- 下記のRTCは、
http://www.openrtp.jp/wiki/_default/ja/Software/SampleRTC.htmlにてダウンロード可能



- このデモで必要だったもの
 - ハードウェア
 - Windows7が動作するPC
 - i-SOBOT
 - Buffalo PC-OP-RS1(USB赤外線リモコン)
 - マイク&スピーカ(今回は、USBスピーカーホンを利用)
 - ソフトウェア
 - OpenRTM-aist-1.0 Python版
 - RTSエディタ
 - AquesTalk1.0 (<http://www.a-quest.com/download/index.html>)
 - Python Serial Port Extension (<http://sourceforge.net/projects/pyserial/>)
 - Pythonでシリアルポートを扱うための拡張
 - Python for Windows Extensions (<http://python.net/crew/skippy/win32/>)
 - PythonでWindowsのAPIを直接コールするための拡張(ダイアログの表示など)
 - 国際ロボット展のRTC([AquesTalkComp.zip](#), [IsobotComp.zip](#))

- Choreonoidの紹介
 - 動力学シミュレーションを行いながら多関節型ロボットの動作を簡単に記述
 - 直感的で簡易なインターフェースを提供
- OpenHRIの紹介
 - OSS版コミュニケーション知能RTC
 - 対話システムを簡単に構成するためのコンポーネント群
 - 音声認識と対話制御の簡素化