

eSEATの使い方と ROSとの連携について

産業技術総合研究所

原 功

- 元々は日本語音声認識Juliusの出力をコマンドに変換するプログラム (SEAT)
 - C言語で実装
 - キーワードマッチングで外部モジュールにメッセージ送信
 - 認識文法切替と対応コマンド列の切替機能



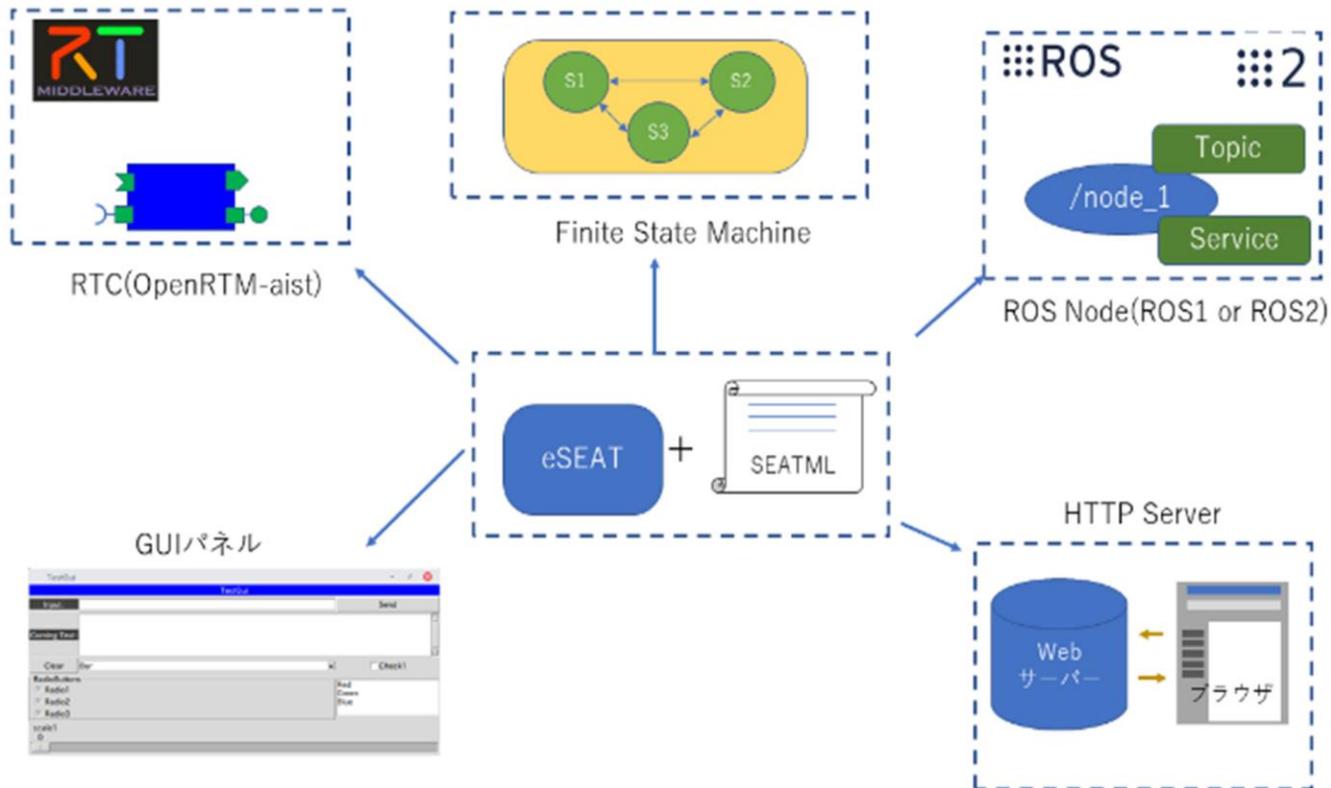
- 松坂要佐氏により、
 - Pythonにより再実装
 - OpenRTM-aistのデータポート追加
 - 状態遷移モデル機能の追加

- 標準的な状態遷移モデルとGUI機能の追加
 - 対話制御の状態モニタリングのためのGUI
 - 標準的な状態遷移モデルの実装



- eSEATに名称変更と高機能化
 - OpenRTM-aistのサービスポート対応
 - CometによるWeb連携機能
 - GUIで使えるアイテムの追加
 - ROSまたはROS2のアダプタの追加

- 簡単な状態遷移マシンを内包したソフトウェアコンポーネント
- "adaptor"を介して外部ソフトウェアとの連携(Web, RTC, ROS Node)
- 簡易GUIパネル(Label, Input, Button, Textarea, Choice, Radio button, Check, Slidebar)



- Windowsでは、[rpt](#)を利用してインストール
- 例：Dドライブへのインストール

Download rpt

```
$ cd /d d:¥
$ git clone https://github.com/haraisao/rtp
```

Update database

```
$ cd rpt
$ rpt update
Download pkgs.yaml:|=====>|(100%)
```

Install packages

```
$ rpt install eSEAT
$ rpt install python-openrtm
$ rpt install OpenRTP
```

- eSEATのソースコードをダウンロードしてインストール

Download eSEAT

```
# git clone git://github.com/haraisao/eSEAT
```

Install packages

```
# cd eSEAT  
# sudo bash ./install.sh
```

/usr/local/eSEATにインストールされる

1. eSEATManagerによるeSEATコンポーネントの生成
2. SEATML_Managerによるseatmlファイルの読み込みと構文解析
 1. generalタグの解釈
 - 各種ポートの生成
 - 初期化スクリプトの実行
 - onExecuteで実行するコード定義
 2. stateタグの解釈
 - stateの生成とルールの定義
3. eSEATコンポーネントの初期化プロセス (onInitialize)
4. eSEATコンポーネントのアクティベーション後の振る舞い
(ポートからの入力と現在のstateのルールに基づく反応処理の実行)

- SETMLの基本
 - General とState

```

<?xml version="1.0" encoding="UTF-8" ?>
<seatml>
  <general>
    - Adaptorの定義<adaptor>
    - 周期実行スクリプト<onexec>
    - 起動時に実行されるスクリプト<script>
    - 大域変数の定義<var>
    - 一つの状態で一定時間入出力がない場合に実行される共通スクリプト(各状態でontimeoutがない場合有効になる)<ontimeout>
    - Active化の時に実行される共通スクリプト(RTCのみ) <onactivated>
    - Deactive化の時に実行される共通スクリプト(RTCのみ) <ondeactivated>
  </general>
  <state name="st1">
    - 状態遷移時に実行するスクリプトなど<onentry> (state内に1つのみ定義可能)
    - 状態を抜けるときに実行するスクリプトなど<onexit> (state内に1つのみ定義可能)
    - 周期実行スクリプト<onexec> (state内に1つのみ定義可能)
    - 一定時間入出力がなかった時に実行するスクリプトなど<ontimeout> (state内に1つのみ定義可能)
    - ruleの定義 <rule>
    - GUI部品の定義 <label, button, input, text, combobox, listbox, radiobutton, scale, frame, labelframe, brk, space>
  </state>
  <state name="st2">
    ...
    ...
  </state>
  ....
  ....
</seatml>
  
```

- ConsoleIn, ConsoleOut
- Adaptorのtype で "rtcout", "rtcin" を使う

```
<adaptor name="out" type="rtcout" datatype="TimedLong" />
```

- コアロジックは、onexecまたはruleで作成

- TalkerとListener
- Adaptorのtype で "ros_pub", "ros_sub" を使う
- Topicはnameで指定する

```
<adaptor name="chatter" type="ros_pub"  
          datatype="std_msgs/String" size="10" />
```

- コアロジックは、onexecまたはruleで作成

- eSEATでは、OpenRTM-aistのデータポートとROSのPublisher/Subscriberの同時利用が可能
- 各Adaptorからのデータ型にあった変換をコアロジック(<script>)で作成する

```
<adaptor name="vel" type="rtcin" datatype="TimedFloatSeq" />
<adaptor name="turtle1/cmd_vel" type="ros_pub" datatype="geometry_msgs.Twist" />
```

```
<rule source="vel">
  <script sendto="turtle1/cmd_vel">
    data=seat.get_in_data()
    cmd=seat.newData("turtle1/cmd_vel")
    cmd.linear.x = data.data[1]/100.0
    cmd.angular.z = -data.data[0]/100.0
    seat.set_result(cmd)
  </script>
</rule>
```

eSEATの詳細については、

http://hara.jpn.com/_default/ja/Software/eSEAT_v2.5.html

をよく読んでください。

「こういう機能はないか？」という問い合わせは、個別に対応します。