



National Institute of
Advanced Industrial Science
and Technology

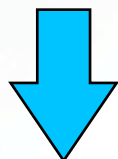
AIST

ロボットへの対話機能付与のための フレームワークSEATとSATの開発

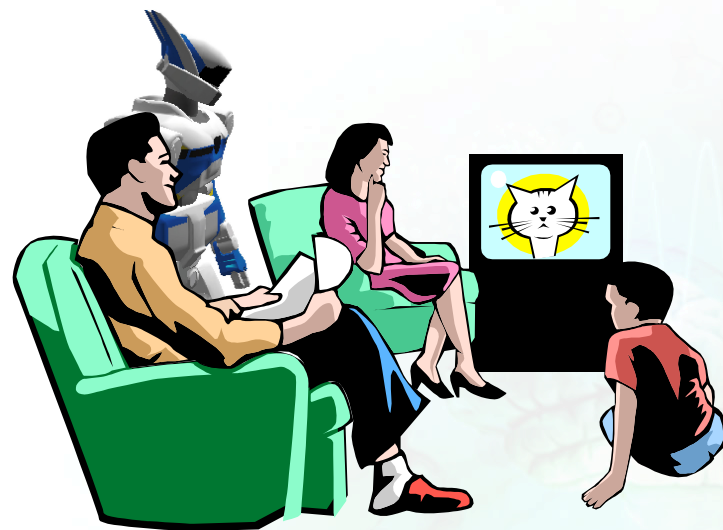
松坂要佐 産業技術総合研究所
藤井洋之 科学技術振興機構
原 功 産業技術総合研究所

研究の背景と目的

- 産業用ロボット
 - 専門知識を持ったオペレータ
 - 決められた行動
- 対人サービスロボット
 - サービス受益者=オペレータ
 - 状況に応じた各種行動



- 知的な意図理解
- コミュニケーション知能



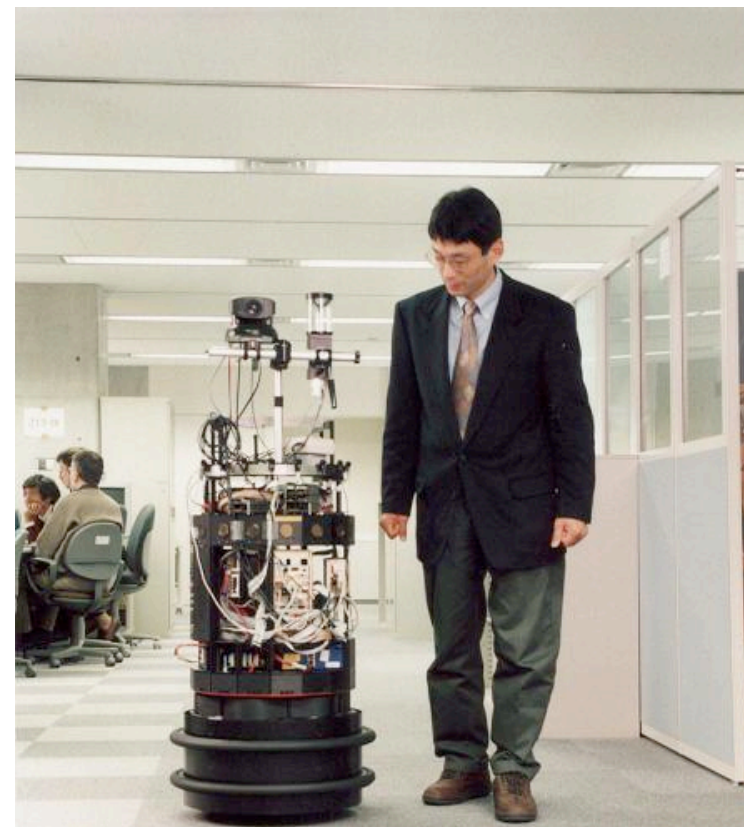
研究の位置づけ

オープンRTプラットフォーム計画 <http://openrtp.jp/>

- ロボットシミュレータ
- ミドルウェアフレームワーク
- 各種知能モジュール

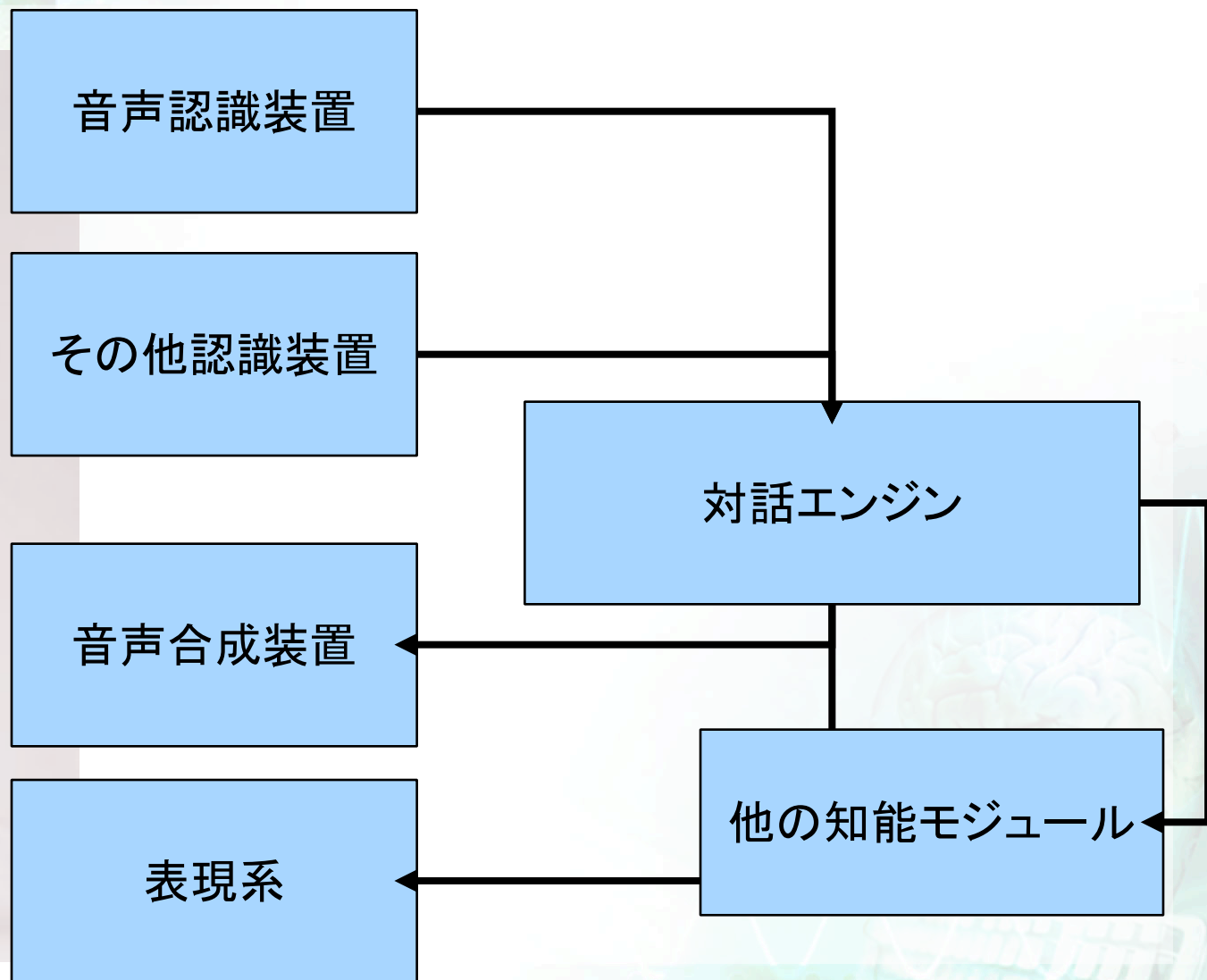
知能化のための基盤技術開発

知能モジュールとしての
対話エンジン開発に関して
NEC, ATR, 九州工業大学...
本エンジンは産総研版



事情通口ロボット1997(電総研)

対話エンジンとは



本ソフトウェアの特徴

- 対話機能の作成を容易化するソフトウェアフレームワーク
 - SEAT(Situated Event Action Transfer)
 - SAT(Situated Action Transducer)
 - 対話の諸問題に対処する各機能
- ロボット用
 - 外部モジュールとのインターフェース
 - 省メモリ性能
- オープンな開発
 - オープンソース
 - 開発ポータル

対話の諸問題 ー対話とは？ー

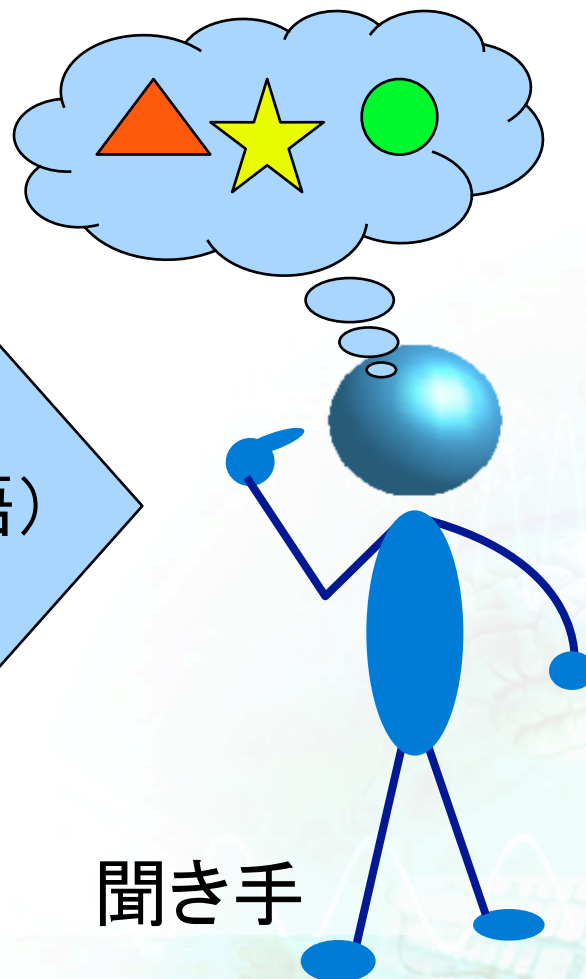
伝えたいコンセプト

=

理解したコンセプト



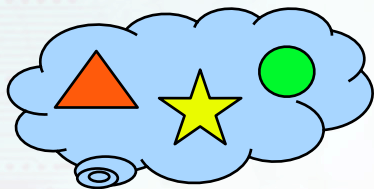
言語(+非言語)



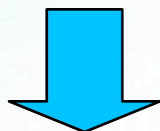
対話の諸問題

発話者(発信者)

伝えたいコンセプト

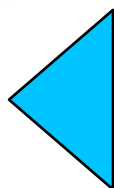


言語化

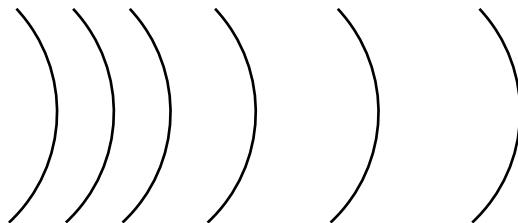


言語表現

発声



空間伝達

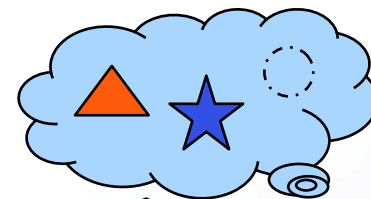


聞き取り

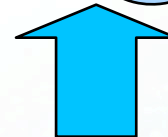


聞き手(受信者)

理解されたコンセプト

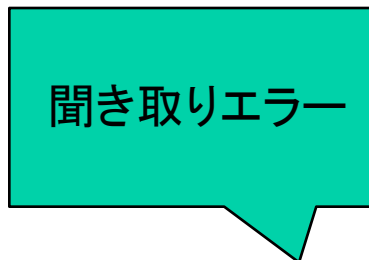


解釈



言語表現

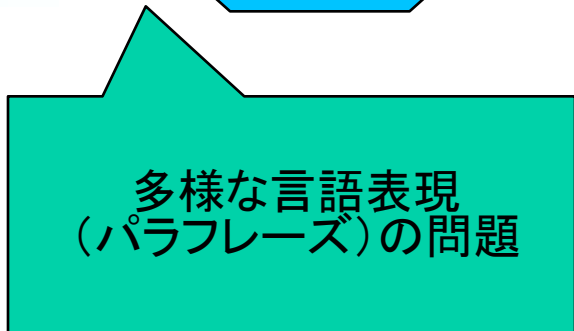
聞き取りエラー



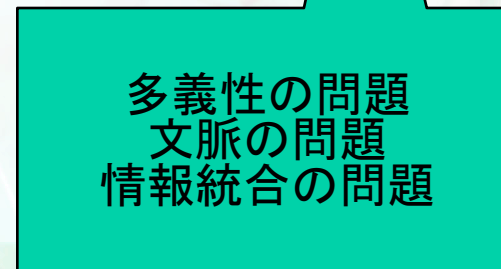
環境ノイズ



多様な言語表現
(パラフレーズ)の問題



多義性の問題
文脈の問題
情報統合の問題



SEAT SATがサポートする機能

多様な言語表現への対応

⇒柔軟なパターンマッチ機能

例: 1歩(前進 | 前へ | 前[に]進んで)

聞き取りエラーの防止

⇒音声認識辞書の切り換え機能

音声認識辞書 = 音素列から言語への変換に関する信念

例: 「あったかい」: 暖かい、あっ高い、あった貝?

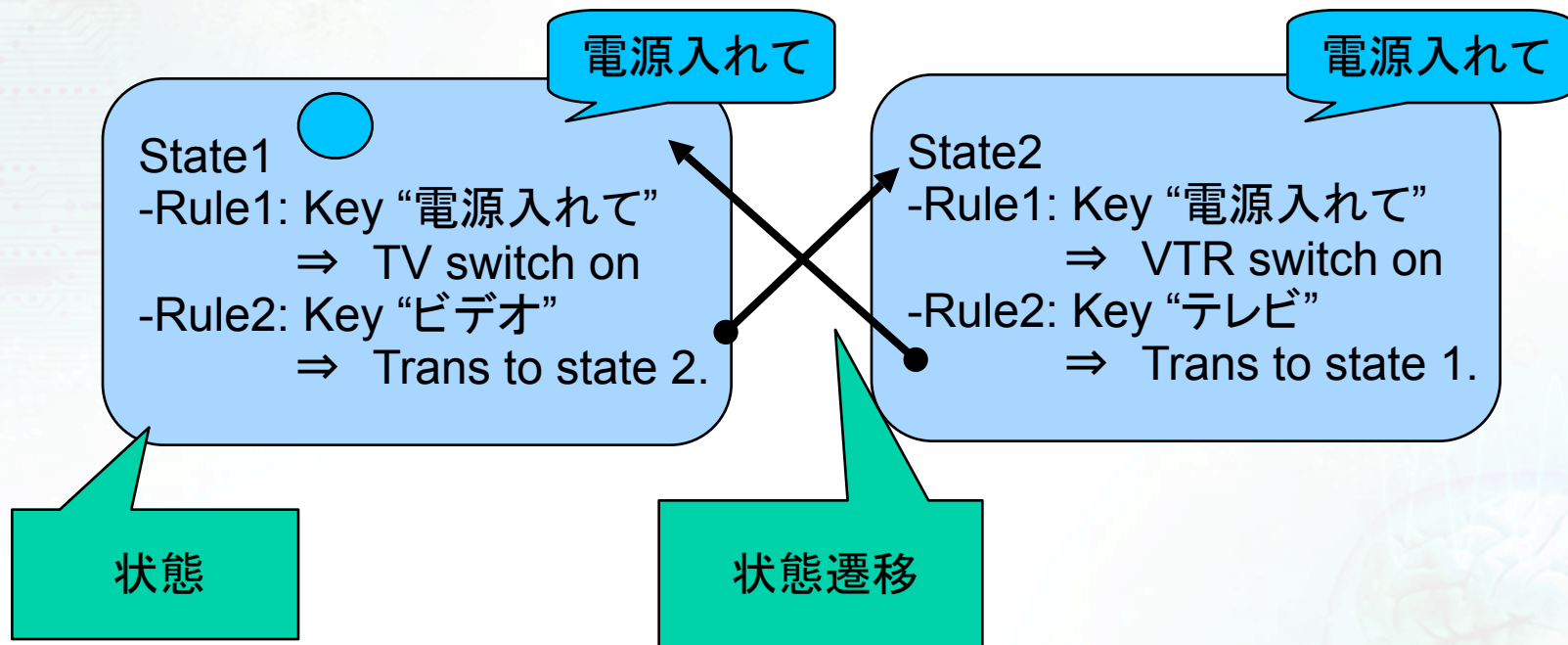
⇒n-bestマッチング機能

複数候補を考慮

例: 「アップ」: 第1候補: カップ、第2候補: アップ

SEATがサポートする機能

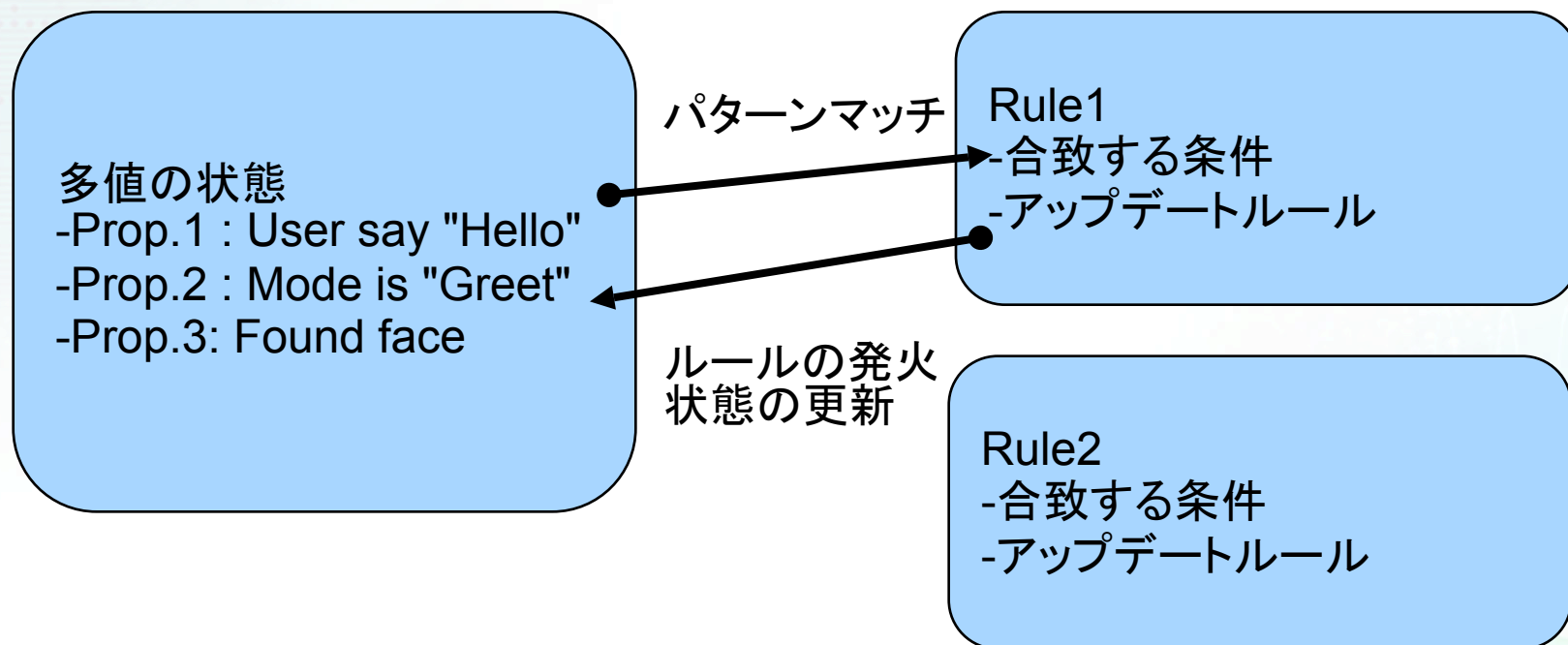
状態遷移モデルに基づく文脈管理機能



既存の様々な研究で使われている文脈管理手法

SATがサポートする機能

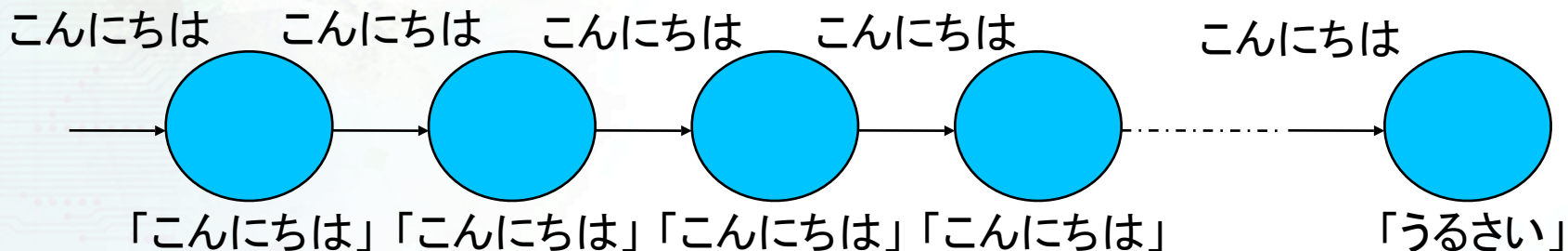
プロダクションシステムモデルに基づく文脈管理機能



複雑な状態を制御できる文脈管理手法

「こんにちは」と10回言われたら「うるさい」

状態遷移モデルによる実現



プロダクションシステムモデルによる実現

ルール1

条件:

ユーザ発話 = 「こんにちは」

繰り返し数 < 10

アクション:

システム発話「こんにちは」

繰り返し数 ++

ルール2

条件:

ユーザ発話 = 「こんにちは」

繰り返し数 ≥ 10

アクション:

システム発話「うるさい」

繰り返し数 = 0

内部状態 (内部状態の変化によって新たなルールを発火 = 演繹)

記述の自由度

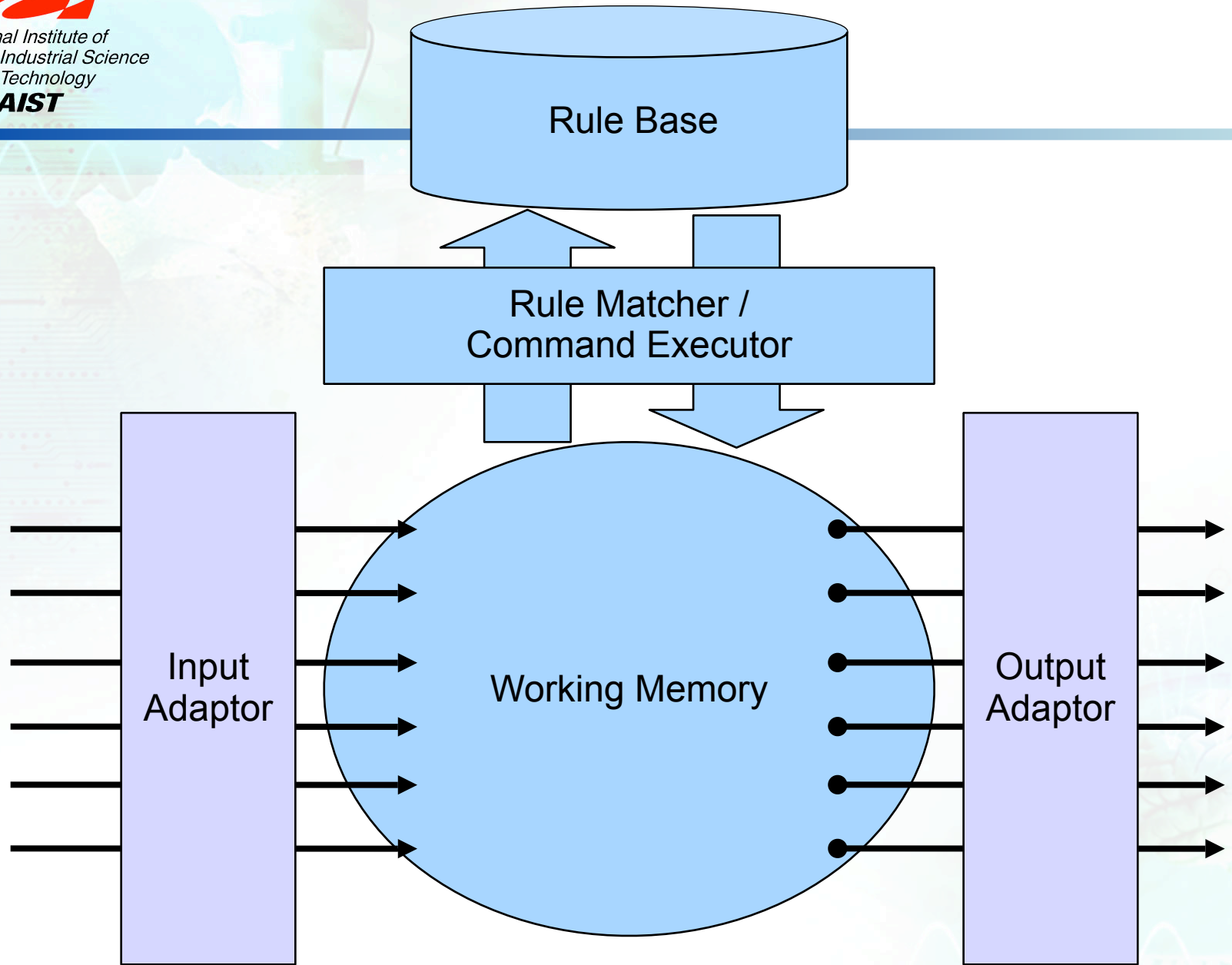
状態遷移モデル <<< プロダクションシステム

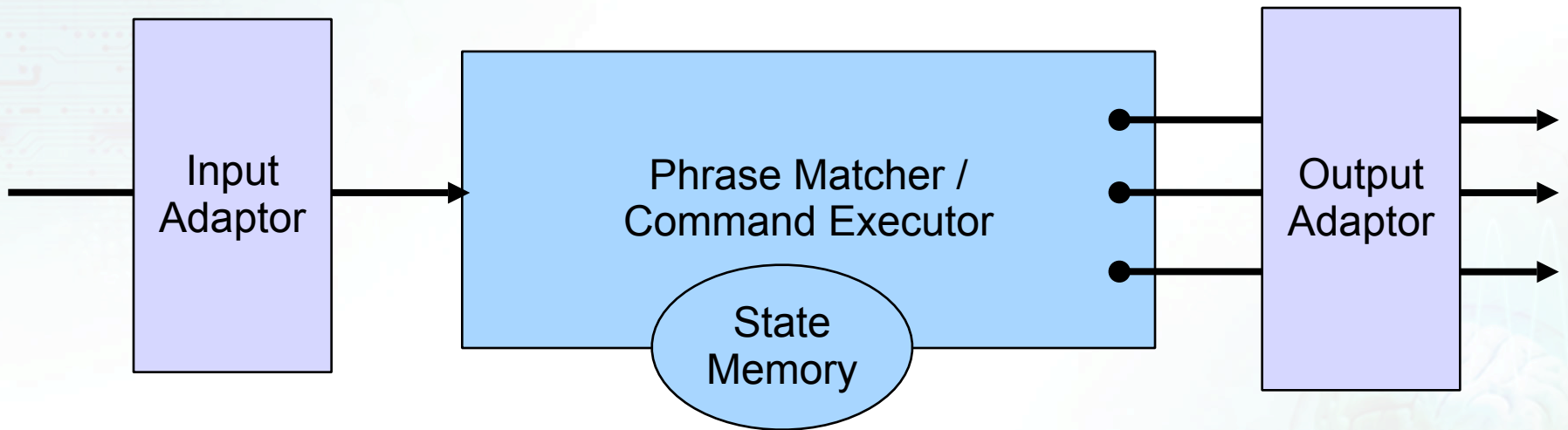
動作の検証可能性

状態遷移モデル >>> プロダクションシステム

両者はトレードオフの関係

⇒タスクに応じてSEAT, SATを選択





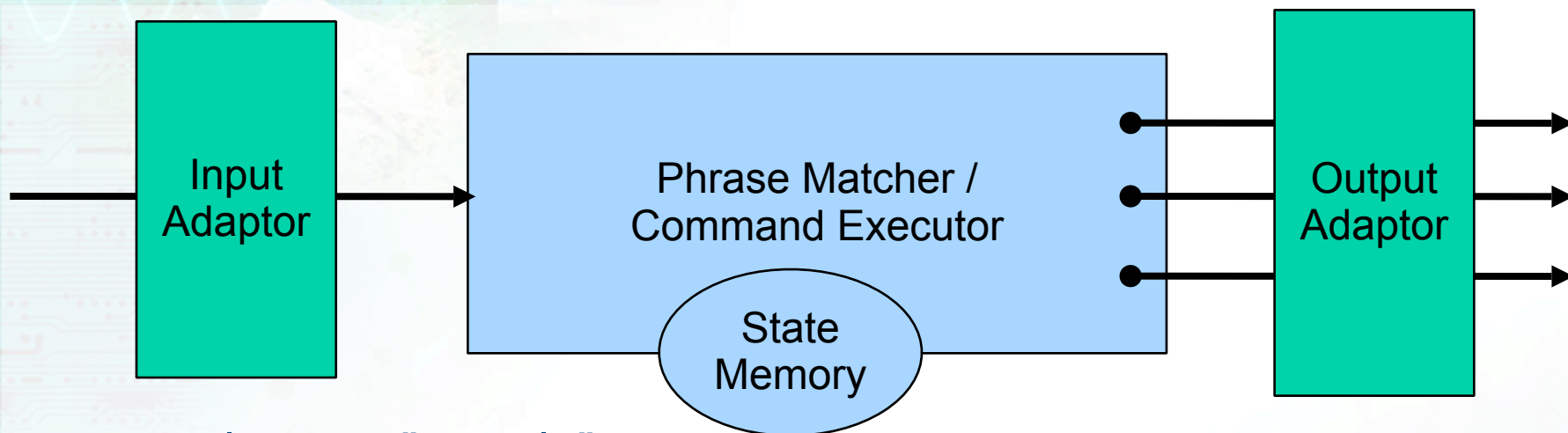
```
<state name="main" dict="juliusdict/  
  main.jgram">  
  <rule>  
    <key>ロボット操作モード[へ変更]</key>  
    <statetransition>  
      robotcontrol  
    </statetransition>  
    <command host="talk">  
      (talk "ロボット操作モードへ変更")  
    </command>  
  </rule>  
</state>
```

```
<state name="robotcontrol"  
  dict="juliusdict/robotcontrol.jgram">  
  <rule>  
    <key>1歩(前進 | 前へ)</key>  
    <key>1歩前に進んで</key>  
    <command host="talk">  
      (talk "1歩前に行きます")  
    </command>  
    <command host="control">  
      (robot hwalk :set-target-pos 0.2 0 0)  
    </command>  
  </rule>  
</state>
```

詳しい記法はwebを参照



アダプタ定義



```
<general name="sample">
```

```
<julius host="localhost" port="9000" code="EUC-JP" />
```

```
<agent name="control" host="localhost" port="10000" />
```

```
<agent name="talk" host="localhost" port="11000" code="shift-jis" />
```

```
<agent name="remote" host="192.168.0.10" port="10000" />
```

```
</general>
```

ソケット通信、音声認識器との通信を抽象化。次バージョンでOpenRTMに対応予定。

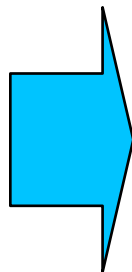
アダプタの差し替え

```
% seat hrp2-adaptor.xml script1.xml script2.xml...  
% seat taizo-adaptor.xml script1.xml  
script2.xml...
```

アダプタ部のみ差し替え

スクリプト部は共有

HRP2(産総研)



たいぞう(JST)

実行ファイルのポータビリティ

ファイルサイズ: 56KByte

実行時メモリサイズ: 約564KByte

実行速度:

SEAT: ハッシュ関数による高速パターンマッチ: $O(1)$

SAT: naive match: $O(N)$ (次バージョンで高速アルゴリズムを実装予定)

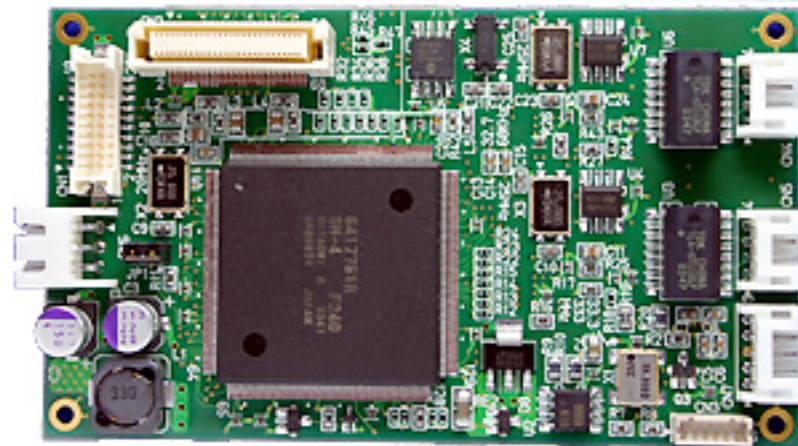
開発言語: C言語 100%

依存ライブラリ:

基本機能はlibcのみで

コンパイル可能

ライセンス: EPL (商用利用も可能)



- ソフトウェアの配布
- 開発したスクリプトの共有機能
(SEAT SATと連携)

[[start]] SEAT,SAT: AIST-OPENRTP PROJECT

現在位置: start

[文書の編集](#) [以前のリビジョン](#) [管理](#) [ユーザー情報の更新](#) [最近の変更](#) [ログアウト](#)

検索 Translations of this page:
[ja](#) [en](#)

SEAT

- SEATについて
- SEATMLの定義
- SEATユーザマニュアル
- 🌐 サンプルスクリプト

SAT

- SATについて
- SATMLの定義
- SATユーザマニュアル

ダウンロードとインストール

- ダウンロード

start

このページでは、ロボット用対話フレームワークSEATとSATについての情報を公開します。

ニュース

- 🌐 プロジェクト作成 (2008/08/18 18:30)
SEATとSATに関するプロジェクトを作成しました。

目次

- start
- ニュース

[編集](#) [編集](#)

ロボット用対話エンジン

- 対話の諸問題に対処する機能
- 柔軟な入出力アダプタ
- オープンソース

Webで公開中

<http://www.openrtp.jp/seatsat/>