オントロジーを利用した人型ロボット Nao による動作を伴う対話の実現

Human Robot Interaction Based on Wikipedia Ontology and Action Ontology

小林 昭太郎　　山口 高平
Shotaro KOBAYASHI　　Takahira YAMAGUCHI

慶應義塾大学
Keio University

This paper discusses how to develop a new system for Human Robot Interaction (HRI) based on Wikipedia ontology and Action ontology. For the purpose, we implement a system which enables robots to communicate with humans in various topics with voice and execute related actions to the topic by relating these ontologies using a fully programmable humanoid robot named Nao. Case studies show us how HRI goes well with two topics: healthcare and movie directors.

1. はじめに

近年，産業用ロボットだけにとどまらず，さまざまな機能を持つロボットが開発されている．その中で，人間と対話することを主たる目的とした対話ロボットも数多く実用化され，普及し始めている．現在，対話ロボットは，介護現場や病院において高齢者の話し相手として利用されている例が数多くある．今後，日本では高齢化がさらに進む事を考慮すると，対話ロボットのニーズがより高まることは疑う余地がない．しかし，現状では，ロボットが知識を持たないため，HRI がスムーズにいかないという大きな問題がある．

このような現状を踏まえ，本稿では，対話モデルと音声インタフェースの既存研究を利用し，そこに体系化された概念定義であるオントロジーを入れることにより，ユーザーとロボット間の音声対話の活性化，および汎用的な対話処理とロボットの行動制御の対応付けの実現を目的とする．この観点から，Wikipedia オントロジーと動作オントロジーに基づいた行動制御可能な音声対話ロボットを開発し，その実証実験の結果について述べる．

2. 提案システム

本稿では，膨大な情報量を有する日本語 Wikipedia から Wikipedia オントロジーを構築し，対話における言語理解用の幅広い辞書として利用する．一方で，ロボット行動制御用のオントロジーとして動作オントロジーを構築し，これらのオントロジーのアライメントをとることにより，ユーザーとロボット間の音声対話の中で，対話内容に関連する動作をロボットに実行させるシステムを提案する．提案システム概要を図1に示す．

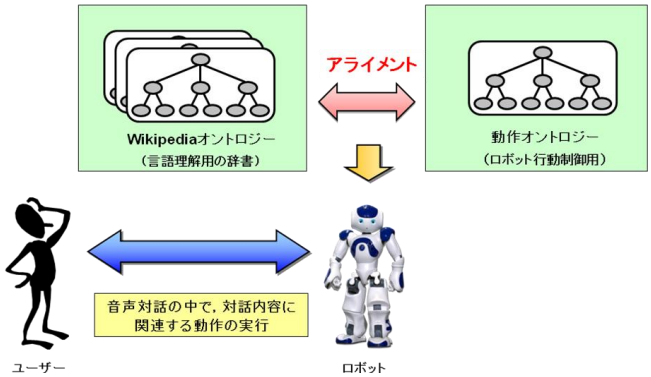

図1　提案システム概要

2.1 使用ロボット Nao

本稿では，Aldebaran 社[*1](仏)製のフルプログラマブル人型ロボット Nao を使用し，全てのシステムの実装を行った．Nao はマイク，スピーカー，LED，センサーなど多くのデバイスを搭載する．また自由度は 25 あり，複雑な動作が実行可能である．さらに，無線によるネットワーク接続が可能で，FTP サーバ機能も有するため，PC からネットワークを通した制御，およびファイルの転送が可能である．なお，システム実装には，Nao のプログラミング言語の1つである Python を利用した．

2.2 Nao の日本語インタフェース

提案システムの構築には，日本語インタフェースが必要不可欠であるが．使用ロボット Nao はこれを持たない．そこで，オープンソースのソフトウェアを組み合わせ，Nao の日本語インタフェースを開発した．

(1) 日本語合成モジュール

日本語合成モジュールの実装には，gtalk[*2]，SoX(Sound eXchange)[*3] の 2 つのソフトウェアを用いる．gtalk は，擬人化エージェントの開発を行う Galatea Project[*4] の構成要素の一つで，日本語テキスト音声合成エンジンである．一方，SoX は音声ファイルに関する様々な機能を有するコマンドラインツールである．これらのソフトウェアは，Nao のオンボードではなく PC 上で起動する．日本語合成モジュールの流れは以下の通りである．まず gtalk と SoX により，日本語テキストから wav ファイルを生成する．生成された wav ファイルを Python の ftplib ライブラリを利用し Nao の FTP サーバにアップロードし，再生する．

ただし，テキストの文字数が多い場合，gtalk で直接合成を行うと，合成処理の所要時間が長くなり，対話に支障をきたす．この問題を解決するために，マルチスレッドによる処理の並列化を行う．具体的には，合成するテキストを句点（。）で区切り，文単位で合成処理を行うことにより，wav ファイルの合成と再生を並列して処理する．これにより，開発した日本語合成モジュールでは，テキストの長さに関わらず，Nao が約1秒程度で話し始めることが可能である．

(2) 日本語認識モジュール

日本語認識モジュールの実装には，Julius[*5]，adintool[*5]の 2 つのオープンソースのソフトウェアを用いる．Julius は大語彙連続音声認識を行うことのできる認識エンジン，adintool は Julius のパッケージに含まれ，音声波形データの記録・分割・送信・受信を行うためのツールである．adintool は Nao のオンボードで起

連絡先：小林昭太郎，山口高平
慶應義塾大学理工学部管理工学科
〒223-8522　神奈川県横浜市港北区日吉 3-14-1
TEL:045-566-1614
E-mail：{s_kobayashi,yamaguti}@ae.keio.ac.jp

*1 http://www.aldebaran-robotics.com/en/
*2 http://sourceforge.jp/projects/galateatalk/
*3 http://sox.sourceforge.net/
*4 http://hil.t.u-tokyo.ac.jp/~galatea/index-jp.html
*5 http://julius.sourceforge.jp/

動し，Julius は PC 上で起動する．日本語認識モジュールの流れは以下の通りである．まず adintool により，Nao に搭載されているマイクが感知した音声波形データをネットワーク経由で PC に送信する．それを Julius で受信し，認識結果を得る．

(3) 言語理解

言語理解にはさまざまな手法が提案されているが，本研究では意味文法により処理する．意味文法を用いた最大の理由としては，実装が容易であることが挙げられる．ただし，意味文法は一般的にDBとの親和性が高いが，オントロジーのクラス，インスタンスはそれぞれ DB のフィールド，データに対応するため，オントロジーとの親和性も高いことも理由の1つである．

3. オントロジーの構築とアライメント手法

3.1 Wikipedia オントロジー

Wikipedia は Infobox やカテゴリ関係など構造化された部分と自由記述によるテキストが共存する半構造化資源である．そのため，Wikipedia オントロジーに基づくドメインオントロジー構築支援環境の実現と評価[桜井 09]の手法を利用し，日本語 Wikipedia から以下の6つの関係を抽出し，構造化されたオントロジーを自動的に構築する．

1. 関連関係(skos:related)
2. シノニム(skos:prefLabel, skos:altLabel)
3. クラス-インスタンス関係（rdf:type）
4. Is-a 関係(rdfs:subClassOf)
5. Infobox トリプル
6. プロパティ定義域（rdfs:domain）

この6つの関係の中で，[3.クラス-インスタンス関係]と[4.Is-a 関係]の2つを，言語理解用の幅広い辞書として利用する．この具体例として，健康法クラスの一部を図2に示す．ただし，図2における色つきの部分がインスタンスを表し，白抜きの部分がクラスを表す．

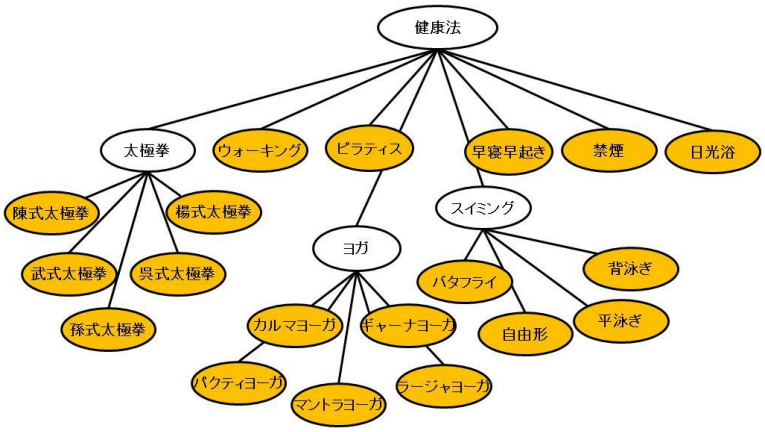

図2 言語理解用の辞書の具体例(健康法クラス)

3.2 動作オントロジー

Nao の動作オントロジーを開発し，さまざまな種類の動作が実行可能な Nao の行動制御のために用いる．

(1) クラス-インスタンス関係

Nao が現時点で実行可能な動作を体系的にまとめ，クラス-インスタンス関係として記述する．図3にこのクラス-インスタンス関係の一部を示す．

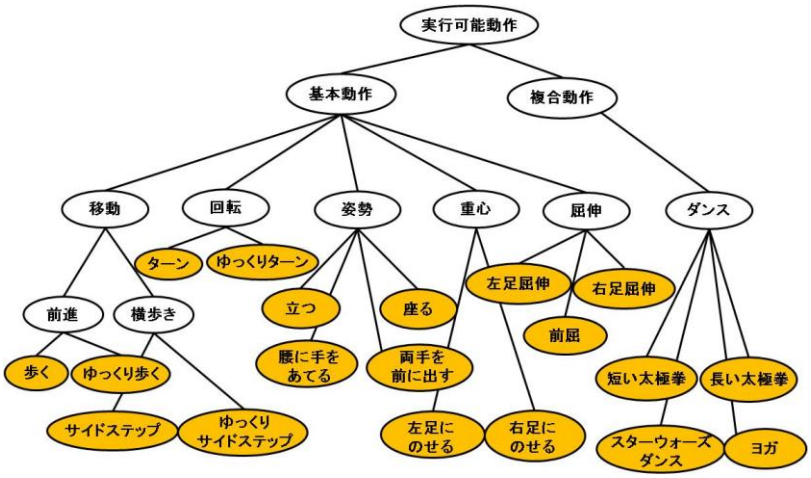

図3 動作オントロジーにおけるクラス-インスタンス関係(抜粋)

図3の色つきの部分がインスタンスであり，Nao の動作実行ソフトウェアモジュールが対応付けられている．一方，白抜きの部分は複数の実行可能動作をまとめるクラスを表す．また，図3から分かるように，全てのインスタンスは，基本動作クラスと複合動作クラスのいずれかに一方に属する．基本動作クラスのインスタンスは，単一の動作ファイル，または Python プログラムによって実行できる動作である．それに対して，複合動作クラスのインスタンスは，複数の基本動作を連続して実行することによって，実行できる動作である．このように実行方法に基づいて，各動作を基本動作と複合動作に分類することによって，動作実行時の処理を明確かつ簡潔に記述することが可能となる．

(2) インスタンスに関する記述

各動作に関するデータは，インスタンスに関する記述として表す．この記述について，基本動作クラス，複合動作クラスそれぞれのインスタンスについて説明する．

- 基本動作クラスのインスタンス

各インスタンスには，難易度と所要時間のデータをそれぞれ Difficulty，Length プロパティを用いて記述する．これらのデータを記述することにより，ユーザーはロボットにある関連動作を実行させるだけでなく，これらのデータに基づき，さらなる動作の要求を行うことが可能になる．例えば，ロボットがある関連動作を実行後，ユーザーがより簡単な動作や，より短時間で終わる動作を要求することができる．さらに，単一の動作ファイルまたはプログラムによって実行される基本動作クラスのインスタンスに関しては，そのファイルのパスを Path プロパティにより記述する．図4に基本動作の1つである「立つ」という動作に対するインスタンスに関する記述を示す．

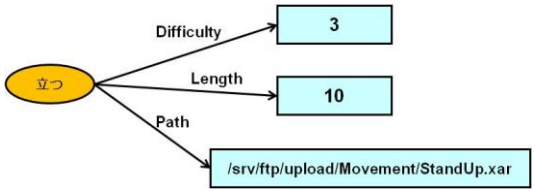

図4 基本動作インスタンスに関する記述（立つ）

- 複合動作クラスのインスタンス

複合動作クラスのインスタンスに関しては，1つの動作ファイルではなく，複数の基本動作を連続して実行することによって実行される．そこで，基本動作クラスのインスタンスに関

する記述における Path プロパティの代わりに，Action_flow プロパティを定義し，RDF のコンテナモデルの１つである rdf:Seq を用いて，複合動作を構成する基本動作フローを記述する．具体的には，Action_flow プロパティの値を rdf:Seq 型のブランクノードとし，rdf:_1，rdf:_2 などのプロパティを用いて，基本動作フローを記述する．また，動作フローの一部を実行することにより，短縮バージョンが実行可能な複合動作に関しては，その実行手順を Short_version プロパティで記述する．図5に複合動作の1つである「短い太極拳」という動作に対するインスタンスに関する記述を示す．

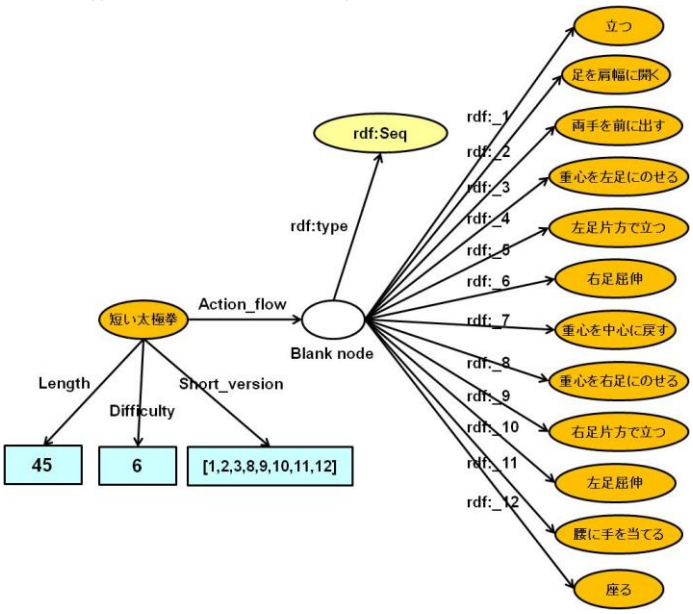

図5　複合動作インスタンスに関する記述（短い太極拳）

3.3 オントロジーのアライメント手法

Wikipedia オントロジーと動作オントロジーは，構築の観点が大きく異なるため，オントロジーマッチングの手法を適用し，自動的にアライメントをとることは困難が伴う．そこで，本稿では意味を考慮して手動でアライメントをとる．具体的には，動作オントロジーのクラス，インスタンスに関連付けたいキーワードを rdfs:label のプロパティ値として手動で登録する手法を提案する．このアライメント手法について，動作オントロジーのインスタンスとクラスそれぞれにキーワードを登録する場合について述べる．

(1)　インスタンスにキーワードを登録

動作オントロジーのインスタンスに rdfs:label を用いてキーワードを登録する場合，それらのインスタンスに，キーワード名と同名の Wikipedia オントロジーのクラス，またはインスタンスに関連付けられる．同名のクラスが存在する場合に関しては，そのサブクラスとスーパークラス，およびそのクラスに属する全てのインスタンスにも関連付けられる．一方，同名のインスタンスが存在する場合については，そのインスタンスが属する全てのクラスにも関連付けられる．このように幅広く関連付けることにより，可能な限り多くの関連動作を検索することができる．

(2)　クラスにキーワードを登録

動作オントロジーのクラスに rdfs:label を用いてキーワードを登録する場合，そのクラスに属する全てのインスタンスに個別にキーワードを登録するのと同等となる．このような枠組みにより，類似する動作に対してまとめキーワードを登録することが可能となり，登録に要するコストを大幅に削減できる．

(1), (2)を踏まえ具体例として，「太極拳」と「ウォーキング」という2つのキーワードの登録に基づく Wikipedia オントロジーと動作オントロジーのアライメントの具体例を図6に示す．

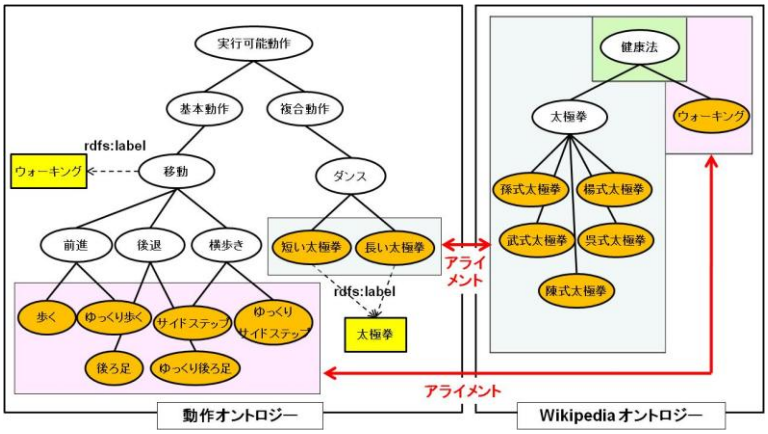

図6　アライメントの具体例

表1に，現時点で動作オントロジーに登録されているキーワードリストの一部を示す．表1における左列の色つき，白抜きは，それぞれ動作オントロジーのインスタンス，クラスを表す．これらのキーワードは，protégé[*6]などのオントロジーエディタを用いることで容易に編集することが可能である．

表　1　登録されたキーワードリスト（抜粋）

動作オントロジー	登録されたキーワード
移動	移動，ウォーキング
ダンス	ダンス，踊り
屈伸	屈伸，ストレッチ，体操
短い太極拳	太極拳，中国武術，中国
長い太極拳	
スターウォーズダンス	映画，スターウォーズ，ジョージルーカス

4.　動作を伴う音声対話ロボットの実現と評価

3.章で示したオントロジー，およびそれらのアライメントに基づく動作を伴う音声対話人型ロボットを実装した．この対話の流れは以下の通りである．まずユーザーが対話のトピックを決定した後，Nao は Wikipedia オントロジーを検索し，その詳細情報を列挙する．次に，ユーザーがこの中の 1 つを選択し，さらなる詳細情報，もしくは関連行動を要求する．ユーザーが詳細情報を要求した場合は，選択された候補について再度 Wikipedia オントロジーを検索し，詳細情報を返す．一方，関連行動の実行を要求した場合については，Wikipedia オントロジーと動作オントロジーのアライメントに基づき，関連動作を検索し，ユーザーに提示する．そして，ユーザーに選択された関連動作を実際に実行する．さらに，インスタンスに関する記述を参照し，関連動作の実行後，ユーザーからの所要時間や難易度に基づく他の動作の要求に応えることも可能である．

4.1 実証実験

構築した動作を伴う音声対話ロボットの実証実験として，健康法，映画監督という2つのトピックにおける対話を行った．それぞれの対話の流れを以下に示す．

*6　http://protege.stanford.edu/

- トピック１： 健康法

(User) 対話のトピックとして，健康法を指定
(Nao) Wikipedia オントロジーの健康法クラスを検索し，その結果を提示
(User) 太極拳を選択し，詳細を要求
(Nao) Wikipedia オントロジーの太極拳クラスを検索し，その結果を提示
(User) 孫式太極拳を選択し，関連動作を要求
(Nao) 動作オントロジーを検索し，関連動作として，短い太極拳と長い太極拳の2つの動作を提示
(User) 短い太極拳を要求
(Nao) 選択された短い太極拳を実行

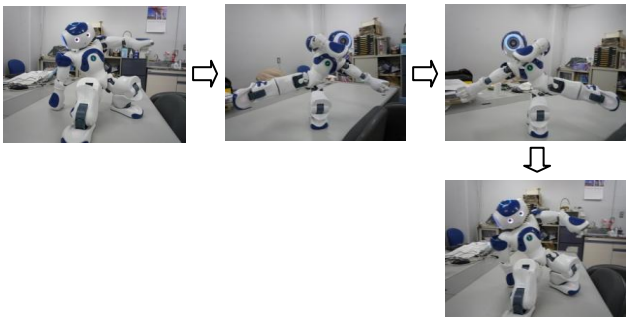

図7 短い太極拳(両足屈伸)の実行

(User) より短時間でできる関連動作を要求
(Nao) 動作オントロジーの Length，および Short_version プロパティを参照し，条件に合う関連動作として，Short_version プロパティに実行手順が記述されている短い太極拳の短縮バージョンを提示
(User) 提示された短い太極拳の短縮バージョンを要求
(Nao) 短い太極拳の短縮バージョンを実行

図8 短い太極拳の短縮バージョン(左足のみ屈伸)の実行

- トピック２： 映画監督

(User) 対話のトピックとして，映画監督を指定
(Nao) Wikipedia オントロジーの映画監督クラスを検索し，その結果を提示
(User) ジョージルーカスの関連動作を要求
(Nao) 動作オントロジーを検索し，関連動作として，スターウォーズダンスを提示
(User) スターウォーズダンスを選択
(Nao) スターウォーズダンスを実行

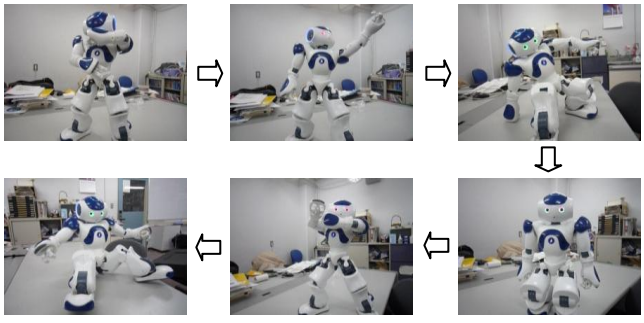

図9 スターウォーズダンスの実行

(User) より難易度の高い関連動作を要求
(Nao) 動作オントロジーの Difficulty プロパティを参照し，条件に合う関連動作が存在しないためユーザーの要求を拒否

4.2 実験結果に関する検討

4.1.章で は健康法と映画監督という2つのトピックにおける動作を伴う対話の流れを示した．これらの結果から明らかであるが，開発した動作を伴う音声対話人型ロボットは，プログラムレベルでトピック毎にシナリオを定義する必要なく，幅広い分野のトピックにおける対話と関連動作の実行が可能である．この点に関しては，既存の音声対話ロボットと比較して優れていると言える．

しかし，このシステムでは，言語理解に意味文法を利用しているため，予め定義した文法パターンに沿った文章，および単語辞書に登録した単語しかユーザーの質問，要求を処理できない．そのため，現状では対話可能なトピックが非常に限られる．したがって，より多くのトピックに対応できるようにしていくことが今後の第一の課題となる．また，現状では，関連動作を全て一度に列挙することしかできないため，多くの関連動作が存在する場合に，ユーザーが実行させたい関連動作を選択するのに非常に時間がかかる．そのため，動作オントロジーの階層関係を利用し，ユーザーが効率的に関連動作を絞り込むことができるフレームワークの構築も今後の課題である．

5. おわりに

本稿では，まずオープンソースのソフトウェアを利用した Nao の日本語インタフェースの構築を行った．さらに，言語理解用の辞書としてのWikipediaオントロジー，ロボットの行動制御のための動作オントロジーを構築し，これらのアライメントをとることにより，動作を伴う音声対話ロボットを開発した．そして，健康法と映画監督という2つのトピックにおける対話実験を行った．これにより，既存の対話モデルと音声インタフェースの研究にオントロジーを入れることにより，音声対話の活性化と，汎用的な対話処理と行動制御の対応付けを実現できることを示した．

今後の展望としては，4.2 章で述べた2つの課題とともに，動作オントロジーの充実に取り組むつもりである．現状では，実行可能動作に対応する動作オントロジーのインスタンス数が少ないため，非常に限られたトピックにおける対話しか関連動作の実行ができない．そのため，新たに動作を設計し，それをインスタンスとして動作オントロジーに組み込むことが必要となる．さらに，動作オントロジーにおけるインスタンスに関する記述に各動作を行う上での前提条件を記述し，そのデータに基づきユーザーの要求に即した動作系列を動的に生成するシステムの構築も必要であると考えている．また，本稿では手動で行った Wikipedia オントロジーと動作オントロジーの自動的なアライメント，さらに多重オントロジーに基づくセマンティックロボットサービスの設計と実現[宮川 09] を応用し，異機能ロボットの連携により，音声対話の中でユーザーの高度な要求を実現する仕組みについても今後検討すべきであると考えられる．

参考文献

[桜井 09] 桜井慎弥，手島拓也，森田武史，和泉憲明，山口高平:Wikipedia オントロジーに基づくドメインオントロジー構築支援環境の実現と評価，第 23 回人工知能学会全国大会 2G1-NFC5-1(2009)

[宮川 09] 宮川智好，山本隆三，植田光，山口高平: 多重オントロジーに基づくセマンティックロボットサービスの設計と実現，第 23 回人工知能学会全国大会 1B2-2 (2009)