

ROSに基づく総合知能アプリケーション開発プラットフォーム PRINTEPSのアーキテクチャ

The Architecture of PRINTEPS: A Total Intelligent Application Development Platform based on ROS

森田 武史*¹
Takeshi Morita

西村 良太*¹
Ryota Nishimura

山口 高平*¹
Takahira Yamaguchi

*¹ 慶應義塾大学大学院理工学研究科
Graduate School of Science and Technology, Keio University

We have developed PRINTEPS (PRactical INTElligent aPplicationS), which is a total intelligent application development platform that integrates 5 types of sub systems (knowledge based reasoning systems, dialog systems, human sensing systems, environment sensing systems, and machine learning systems). This paper proposes the architecture of PRINTEPS based on the Robot Operating System (ROS) and describes a workflow editor for end users and evaluation metrics for PRINTEPS.

1. はじめに

現在、知識推論、音声対話、人と物体の画像センシング、動作という4種類の要素知能(バックグラウンドで機械学習)を統合した総合知能アプリケーション開発プラットフォームPRINTEPS(PRACTICAL INTELLIGENT APPLICATIONS)の研究を進めている[山口15]。PRINTEPSは、開発者向けではなく、エンドユーザが設計段階から参加し(ユーザ参加型デザイン)、ソフトウェア(SW)モジュール結合により知能アプリケーションを容易に(数時間から数日で)開発できることを目指している。

[森田15]で提案したアーキテクチャは、セマンティックWebサービスを基礎としていたが、リアルタイムなデータ処理や異なるシステム間の入出力データ変換が困難であるという問題があった。本稿では、ROS(Robot Operating System)[Quigley09]を用いることにより、これらの問題を解決可能なアーキテクチャを提案する。また、エンドユーザ向けのインタフェースとして開発を進めている、PRINTEPSにおけるワークフローエディタについて述べる。

PRINTEPSは、人と機械のハーモニアス(調和的)な創造過程でえられる体験共有にも対応できる知能共進化を考えている。そのため、PRINTEPSを評価するには、ソフトウェア工学で検討されてきた評価指標(実装容易性、再利用性)の拡張だけでなく、知能に関わる新しい評価指標(多様性、知能共進性)も必要である。本稿では、これらの4つの評価指標についても述べる。

PRINTEPSのケーススタディとして、ロボット喫茶店とクラスルームAIに取り組んでいるが、これらのケーススタディは、別稿[菅原16, 西村16, 眞鍋16, 石井16, 菅16]で述べる。

2. 関連研究

人工知能・知能ロボットアプリケーションの研究では、記号概念レベルの情報と物理信号レベルの情報を汎用性に統合する方法が、長年、重要な研究課題になっている。例えば、ブレーメン大学(独)で推進されているKnowRobプロジェクト[Tenorth13]では、オントロジー記述言語OWL-DLによ

連絡先: 森田武史, 慶應義塾大学理工学部, 〒223-8522
神奈川県横浜市港北区日吉 3-14-1, 045-566-1599,
t.morita@ae.keio.ac.jp

表 1: 各知能の入出力と時間スケール

知能	入出力	時間スケール
知識推論	SWRL, OWL, RDF	一定時間
音声対話	XML	準リアルタイム
感情処理	XML	準リアルタイム
人・環境センシング	ROS	リアルタイム
動作	ROS	リアルタイム

り行動オントロジーが記述され、Prologにより行動計画が実行され、物理信号を処理する各種プログラムが連携されている。IROS国際会議では、2010年より、ロボット分野に特化したDSL(Domain-Specific Language, ドメイン特化言語)のワークショップが開催され、開発者ではなくユーザがロボットを操作できることを目指している。しかしながら、これらの研究は、主に、ロボットの自律的動作である動作知能を対象としている。近年、よく取り上げられる自動運転車やクイズ人工知能ワトソンなども、特定知能であり、極めて高い性能をもつAIアプリケーションであるが、応用範囲は限定されている。すなわち、これらの研究は、本研究が目指している、記号から信号までの複数の知能を連携して、総合的な人工知能・知能ロボットアプリケーションを開発できるものではない。

3. PRINTEPS

3.1 記号と信号の統合における課題

[森田15]で提案したアーキテクチャでは、セマンティックWebサービスの一つであるOWL-S^{*1}を基礎として、最小機能であるモジュールをWebサービスとして実装することを想定していた。しかしながら、表1に示すように、本研究が対象とする総合知能アプリケーションでは、知能ごとに入出力データ構造や処理を行う際の時間スケールが異なっている。入出力データ構造をセマンティックWeb標準のRDF^{*2}形式で統一することは、既存の知識推論以外のライブラリの再利用や新規開発を考慮した場合に、困難であった。また、Webサービスは、知識推論処理のように一定時間(数秒程度)で処理が行えれば良い場合には有用であるが、準リアルタイム(音声対話、感情処理)やリアルタイム処理(人・環境センシング、動作)

*1 <https://www.w3.org/Submission/OWL-S/>

*2 <http://www.w3.org/RDF/>

のように、数十から数百ミリ秒単位で処理が必要となるような場合には、ネットワーク遅延などにより、リアルタイムに処理が行えないという問題があった。

以上の問題を解決するために、本稿では、[森田 15] での問題点を解決可能な ROS に基づく PRINTEPS のアーキテクチャを提案する。

3.2 PRINTEPS における ROS の利用

ROS [Quigley 09] は、オープンソース・ロボティクス財団により開発されている、ロボットフレームワークであり、通信ライブラリや様々なツール群を提供している。ROS では、Web サービスと同様の同期通信の仕組みとしてサービスが利用可能であり、さらに、非同期通信の仕組みとして、トピックも利用可能となっている。ROS は分散システムとして実装されているため、マルチロボットやマルチセンサーを容易に扱うことができ、トピックを用いることにより、リアルタイムにセンサーの値を取得するなどの処理が可能となっている。また、サービスとトピックの入出力データ型は、メッセージとしてクラスや構造体のような複合データ型として定義可能であり、サービスについては、srv ファイルにより、Web サービスにおける WSDL(Web Service Description Language)^{*3} のように、プログラミング言語非依存の形で入出力データ仕様が定義可能となっている。現在、ROS の情報をまとめている Wiki には、2,000 以上の ROS のパッケージが公開されており^{*4}、画像、センサー、動作などに関する様々なライブラリが、ROS のトピックまたはサービスの仕様に従って、世界中の研究者を中心に活発に開発されている。

以上より、PRINTEPS では、最小機能であるモジュールとして、ROS におけるサービスおよびトピックを利用することとした。これにより、豊富な既存ライブラリが再利用できると共に、知能ごとの入出力データ構造が異なる問題や、リアルタイム処理については、トピックを用いることにより解決ができる。

現状の ROS では、パッケージのリストを Wiki で提供しているのみであり、ユーザが作成したいアプリケーションに利用可能な ROS サービスやトピックを検索することは容易ではない。ユーザは関連論文や個別のパッケージのドキュメントやソースコードを読むことにより、必要な ROS サービスやトピックを検索している。一方、PRINTEPS では、SOA に基づいて、ワークフロー形式で、ROS サービスやトピックをモジュールとして構成したプロセスやサービスを構築し、それらをユーザ間で共有することにより、ROS サービスやトピックがどのようなアプリケーション、サービス、プロセスで利用されているかが明示化される。これにより、モジュールの検索が容易となり、モジュールの再利用性も高まることが期待できる。

3.3 ROS に基づく PRINTEPS のアーキテクチャ

図 1 に、ROS に基づく PRINTEPS のアーキテクチャ構成図を示す。PRINTEPS は、ROS に準拠したモジュールの入出力仕様および通信プロトコルを実装した、5 つのサブシステム（知識推論システム、音声対話システム、画像センシングシステム、動作管理システム、機械学習システム）から構成される。知識推論システムは、ワークフロー、ビジネスルール、オントロジーを主に参照し、音声対話システムは、応答生成用ルール、対話処理用ルール、言語理解ルールを参照する。知識推論システムの詳細については、[菅原 16, 石井 16, 菅 16] を、音声対話システムの詳細については、[西村 16, 眞鍋 16]

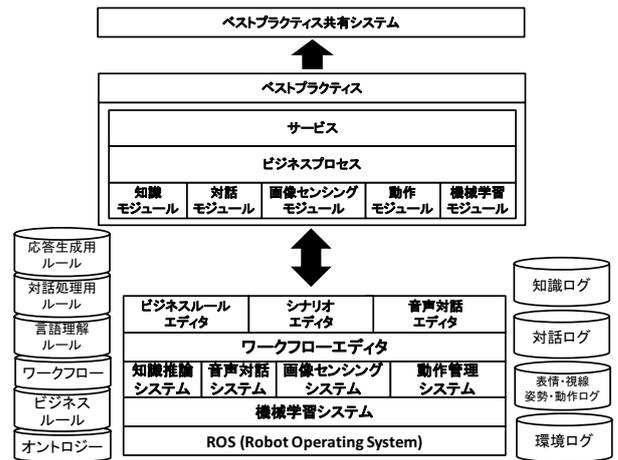


図 1: ROS に基づく PRINTEPS のアーキテクチャ構成図

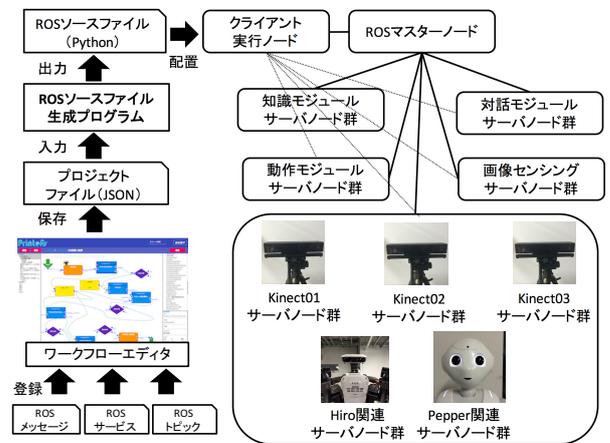


図 2: ワークフローエディタと ROS の関連図

を参照いただきたい。また、人間と機械のマルチモーダルなインタラクションにより取得した知識ログ、対話ログ、表情・視線・姿勢・動作ログ、環境ログなどは、オンラインおよびオフラインの機械学習の入力となり、機械学習モジュール作成に利用される。

PRINTEPS は、AI アプリケーションをユーザ自身が容易に開発できるように、ビジネスルールエディタ、シナリオエディタ、音声対話エディタ、ワークフローエディタを提供する。本稿では、ワークフローエディタを中心に説明する。

ワークフローエディタでは、SOA (サービス指向アーキテクチャ) に基づき、知識、対話、画像センシング、動作、機械学習の各種モジュール群を組み合わせ、ビジネスプロセスのワークフローを構成し、ビジネスプロセスを組み合わせ、サービスのワークフローを構成し、サービスのワークフローにより実現されるアプリケーションをベストプラクティスとして構築できる。構築されたベストプラクティスは、ベストプラクティス共有システムにより、ユーザ間で共有される。

4. ワークフローエディタ

現在、PRINTEPS におけるワークフローエディタを設計し、Web アプリケーションとして実装を進めている。

図 2 に、ワークフローエディタと ROS の関連図を示す。ワー

*3 <https://www.w3.org/TR/wsdl>

*4 <http://www.ros.org/browse/list.php>

表 2: ワークフローエディタの基本構成要素

カテゴリ	説明	図
コネクタ	処理フロー	
	データフロー	
構成要素	サービス	
	プロセス	
	モジュール(ROSトピック)	
	モジュール(ROSサービス)	
	モジュール(メッセージ)	
基本制御部品	条件分岐	
データ操作処理	データ蓄積参照	
	固定データ	
その他	ルート開始	
	ルート終了	
	入力	
	出力	

ワークフローエディタでは、ROSにおけるメッセージ、サービス、トピックに基づいてモジュールが開発されていることを前提としている。図2では、ROS マスターノードに、知識、対話、動作、画像センシング（具体例として、Kinect、Hiro、Pepper）モジュールのノード群が接続されていることを示している。このように、PRINTEPSでは、ROSのサービスサーバやトピックのノードが、分散環境でROS マスターノードに接続された状態で起動している際に、それらをワークフローに従って呼び出すプログラムを自動生成することができる。

ワークフローエディタは、ユーザがSOAに基づいてワークフローを構築した後に、ワークフローにおけるサービスとプロセスおよびプロセスとモジュールの包含関係、モジュール間の入出力関係などを保存したプロジェクトファイル（JSON）を保存可能である。プロジェクトファイルを入力として、ROSソースファイル生成プログラムにより、ROS環境で実行可能なROSソースファイル（Python）を生成する。その後、ROS環境にソースファイルを配置し、ROS マスターノードと接続することで、クライアント実行ノードとして実行することが可能である。

表2に、ワークフローエディタの基本構成要素を示す。ワークフローエディタでは、ROSにおけるサービス（サーバとクライアント）、トピック（パブリッシャとサブスクリバ）、メッセージ（分解と合成）を最小機能であるモジュールとして参照可能となっている。モジュール間の接続については、ROSのメッセージ型に基づいて、自動的に接続可能性を判定可能にする。メッセージモジュールは複合データ型における特定のフィールドの値を別のモジュールの入力と接続する際の型変換に用いることができ、ROSのメッセージ定義ファイルを登録すると、自動的に分解と合成モジュールが作成される。メッセージモジュール（分解）により、複合データ型からその各フィールドを、メッセージモジュール（合成）により、各フィールドから複合データ型への型変換を行うことができる。

プロセスは、複数のモジュールをまとめるための関数的な役割を持ち、プロセス内には、プロセスおよびモジュールを混在して定義することができる。サービスは、最も粒度の粗い機能であり、プロセスのみから構成される。サービスに包含されるプロセスはビジネスプロセスとして定義する。ワークフロー

エディタにおける第1階層（ルートサービス）は、ルート開始、サービスリスト、ルート終了のみから構成される。完成したサービスリストは、ベストプラクティスとして、ユーザ間で共有できるようにする。これにより、どのようなサービス、プロセス、モジュールがベストプラクティスで利用されているかを容易に検索できる。

その他に、基本制御部品、グローバルおよびローカルなデータ蓄積・参照部品、ユーザが直接入力するための固定データ部品なども提供する。自動生成されたソースコード実行時にエラーが発生した場合には、ROS コンソールにトレースメッセージを表示すると共に、どのサービスやプロセス内のモジュールがエラーの原因であるかを特定することができるように、IDを割り振る。

さらに、各モジュール、プロセス、サービスには、入出力データ型、説明文、アクタとそのアイコンなどのメタデータを定義することができ、様々な観点から検索を可能にする。

図3に、ワークフローエディタのスクリーンショットを示す。

5. 評価指標

PRINTEPSを評価するために、実装容易性、再利用性、多様性、知能共進性の4種類の評価指標の具体化と検証を行う。

実装容易性とは、実装時間に関する評価尺度である。例えば、PRINTEPSの開発事例であるロボット喫茶店において、双腕ロボットが食器を把持するタスクに対して、スキルの高い学生が直接コーディングした場合、実装時間は1時間程度であったが、スキルの低い学生でもPRINTEPSを使うと20分程度で同タスクを実装でき、開発時間が約1/3程度になった[石井16]。この予備実験から、ITスキルの低いユーザでも、PRINTEPSを使えば、ITスキルの高い専門家による実装にかかる時間と比較して、1/3程度の時間で開発できることを目標とする。

再利用性とは、SWモジュール群の再利用性である。ロボット喫茶店の場合、喫茶店のレイアウト、接客ルールなどの変化、小学校教諭とロボット連携による授業実践の場合、教諭と授業科目と学習単元などの変化に対して、PRINTEPSのSWモジュール群がどの程度再利用可能であるかを評価する。再利用性が高まれば、その結果、知能アプリケーションの開発時間が短縮される。例えば、後者に関しては、2016年1月、慶應義塾幼稚舎6年生に実施した理科実験授業「ロボットと学ぼう：左右のつりあい～てこの原理の実験～」において、PRINTEPSの開発以前であったため、2時間前後の打合せを8回実施し、知能アプリケーション開発時間は300時間を超えた[菅16]。今後は、PRINTEPSにより、SWモジュール群の再利用を計りながら、打合せ回数を半分、開発時間を100時間未満に短縮する事を目標とする。

多様性とは、開発できる知能アプリケーションの多様性である。例えば、ロボット喫茶店を発展させたロボットレストランにおいて、「注文と食品成分表を連携したアレルギー注意喚起ができる」、「食事レシピと調理動作を連携した細かい注文（固めや柔らかめなど）に対応できる」といった知的機能を創案し、要素知能アプリの連携によるそれらの知的機能を実装することにより、多様性を具体化していく。

知能共進性とは、人と知能アプリケーションの相互作用に基づく知のPDCAサイクルを通して外在化されてくる新しい知識量である。例えば、小学校教諭とマルチロボットの協働による授業実践において、学習単元のまとめ方、演習問題の答え合わせの方法等に関する「教諭とロボットの有効な連携方法」に

